Proceedings of the 2007 IEEE Symposium on Approximate
Dynamic Programming and Reinforcement Learning (ADPRL 2007)

Discrete-time nonlinear HJB solution using Approximate dynamic programming: Convergence Proof

Asma Al-Tamimi, Frank Lewis

***Abstract*— In this paper, a greedy iteration scheme based on approximate dynamic programming (ADP), namely Heuristic Dynamic Programming (HDP), is used to solve for the value function of the Hamilton Jacobi Bellman equation (HJB) that appears in discrete-time (DT) nonlinear optimal control. Two neural networks are used- one to approximate the value function and one to approximate the optrimal control action. The importance of ADP is that it allows one to solve the HJB equation for general nonlinear discrete-time systems by using a neural network to approximate the value function. The importance of this paper is that the proof of convergence of the HDP iteration scheme is provided using rigorous methods for general discrete-time nonlinear systems with continuous state and action spaces. Two examples are provided in this paper. The first example is a linear system, where ADP is found to converge to the correct solution of the Algebraic Riccati equation (ARE). The second example considers a nonlinear control system.**

Key words: Adaptive critics; Approximate dynamic programming; HJB, Policy iterations.

I. INTRODUCTION

This paper is concerned with the application of approximate dynamic programming techniques (ADP) to find the value function of the DT HJB that appears in optimal control problems. ADP is an approach to solve dynamical programming problems utilizing function approximation. ADP was proposed by Werbos [12], Barto *et. al.* [7], Widrow *et. al.* [21], Howard [13], Watkins [10], Bertsekas and Tsitsiklis [17], and others as a way to solve optimal control problems forward-in-time. Therefore ADP combines adaptive critics, a reinforcement learning technique, with dynamic programming.

Several approximate dynamic programming schemes appear in literature. Howard [13] proposed iterations in the policy space in the framework of stochastic decision theory. In [1], Bradtke *et al.* implemented a Q-learning policy iteration method for discrete-time linear quadratic optimal control problems. Hagen [5] discussed the relation between the Q-learning method and model-based adaptive control with system identification. Policy iteration methods for continuous-time optimal control were given in [19][20].

Werbos [14] classified approximate dynamic programming approaches into four main schemes: Heuristic Dynamic Programming (HDP), Dual Heuristic Dynamic Programming (DHP), Action Dependent Heuristic Dynamic Programming (ADHDP), also known as Q-learning [10], and Action Dependent Dual Heuristic Dynamic Programming (ADDHP). In [16], Prokhorov and Wunsch developed new approximate dynamic programming schemes known as Globalized-DHP (GDHP) and ADGDHP. Landelius [8] applied HDP, DHP, ADHDP and ADDHP techniques to the discrete-time linear quadratic optimal control problem and discussed their convergence, showing that they are equivalent to iterating on the underlying Algebraic Riccati equation. In [30], policy iterations are implemented on a second order representation of the original DT HJB equation. In our previous work, we developed an ADP technique to solve the dynamic programming problems encountered in zero-sum games related to the H-infinity control problems of linear systems [2][3]. The current status of work on approximate dynamic programming is given in [4]. See also [17], [29], [27] and [28] for general adaptive critic methods.

Solutions to the DT HJB equation with continuous state space and action space have appeared in [31] where a Taylor series expansion of the Value function is derived. Policy iteration schemes require an initially stable policy. In this paper, we use the greedy HDP iteration scheme, which does not require an initially stable policy. The greedy iteration ADP scheme presented in this paper is applied to solve the DT HJB of the optimal control problem for general nonlinear discrete –time systems.

The importance of this paper is that we provide a rigorous proof of convergence of the HDP algorithm that solves for the value function of the DT HJB appearing in discrete-time nonlinear optimal control problems. Next in

Research supported by the National Science Foundation ECS-0501451, the Army Research Office W91NF-05-1-0314

Corresponding author: A. AL-Tamimi is with the Automation & Robotics Research Institute, The University of Texas at Arlington, Fort Worth, TX 76118 USA (e-mail: altamimi@arri.uta.edu).

F. L. Lewis is with the Automation & Robotics Research Institute, The University of Texas at Arlington, Fort Worth, TX 76118 USA (e-mail: lewis@uta.edu).

the paper, neural network parametric structures are used to approximate the optimal policy and value function of the DT HJB. As is known, this provides a procedure for implementing the HDP algorithm. The paper ends with two examples that show the practical effectiveness of the ADP techniques.

II. The Discrete-Time HJB Equation

Consider an affine in input nonlinear dynamical-system of the form

$$x_{k+1} = f(x_k) + g(x_k)u(x_k) \tag{1}$$

where $x \in \mathbb{R}^n$, $f(x) \in \mathbb{R}^n$, $g(x) \in \mathbb{R}^{n\times m}$ and the input $u \in \mathbb{R}^m$. Assume that the system (1) is stabilizable on $\Omega \in \mathbb{R}^n$. It is desired to find $u(x_k)$ which minimize the cost function given as

$$V(x_k) = \sum_{i=k}^{\infty} x_i^T Q x_i + u_i^T R u_i \tag{2}$$

where $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are positive definite, *i.e.* $\forall x \neq 0$ $x_k^T Q x_k > 0$ and $x = 0 \Rightarrow x^T Q x = 0$. Hence, the class of controllers need to be stable and guarantee that (2) is finite, *i.e.* admissible controls [20].

Definition 1 Admissible Control. A control $u(x)$ is defined to be admissible with respect to (2) on Ω if $u(x)$ is continuous on Ω, $u(0) = 0$, u stabilizes (1) on Ω, and $\forall x_0 \in \Omega$ $V(x_0)$ is finite.

Equation (2) can be written as

$$\begin{aligned} V(x_k) &= x_k^T Q x_k + u_k^T R u_k + \sum_{i=k+1}^{\infty} x_i^T Q x_i + u_i^T R u_i \\ &= x_k^T Q x_k + u_k^T R u_k + V(x_{k+1}) \end{aligned} \tag{3}$$

From Bellman optimality principle, the HJB equation comes out to be

$$V^*(x_k) = \min_{u_k}(x_k^T Q x_k + u_k^T R u_k + V^*(x_{k+1})) \tag{4}$$

The optimal control u^* satisfies the first order necessity condition for the gradient of right hand side of (4) with respect to u

$$\frac{\partial V^*(x_k)}{\partial u_k} = \frac{\partial (x_k^T Q x_k + u_k^T \; R u_k)}{\partial u_k} + \frac{\partial x_{k+1}}{\partial u_k}\frac{\partial V^*(x_{k+1})}{\partial x_{k+1}} = 0 \tag{5}$$

and therefore

$$u^*(x_k) = \frac{1}{2} R^{-1} g(x_k)^T \frac{\partial V^*(x_{k+1})}{\partial x_{k+1}} \tag{6}$$

Substituting (6) in (4) one obtains the DT HJB, where V^* is the value function of the optimal control u^*.

$$\begin{aligned} &V^*(x_k) = x_k^T Q x_k + \\ &\frac{1}{4}\frac{\partial V^{*T}(x_{k+1})}{x_{k+1}} g(x_k) R^{-1} g(x_k)^T \frac{\partial V^*(x_{k+1})}{x_{k+1}} + V^*(x_{k+1}) \end{aligned} \tag{7}$$

In the next section we apply the HDP algorithm to solve the value function V^* of the HJB equation (7) and present a convergence proof of this algorithm.

III. The HDP Algorthim and its Convergence

This section is organized as follows. In the first subsection, the derivation of the HDP algorithm is given, then in the second subsection a proof of convergence of the HDP algorithm is presented for the first time, and finally the last subsection shows how to implement the HDP algorithm with parametric structures like neural networks.

A. *Derivation of the algorithm*

In the HDP algorithm, one starts with initial cost function $V_0(x) = 0$ which is not necessary the value function, and then finds the control u_0 as follows

$$u_o(x_k) = \arg\min_u (x_k^T Q x_k + u^T R u + V_0(x_{k+1})) \tag{8}$$

then one updates the cost as

$$\begin{aligned} V_1 &= x_k^T Q x_k + u_0^T(x_k) R u_0(x_k) + V_0(f(x_k) + g(x_k)u_0(x_k)) \\ &= x_k^T Q x_k + u_0^T(x_k) R u_0(x_k) + V_0(x_{k+1}) \end{aligned} \tag{9}$$

The HDP scheme therefore iterates between

$$u_i(x_k) = \arg\min_u (x_k^T Q x_k + u^T R u + V_i(x_{k+1})) \tag{10}$$

$$\begin{aligned} V_{i+1} &= \min_u (x_k^T Q x_k + u^T R u + V_i(x_{k+1})) \\ &= x_k^T Q x_k + u_i^T(x_k) R u_i(x_k) + V_i(f(x_k) + g(x_k)u_i(x_k)) \end{aligned} \tag{11}$$

In Figure 1, the flow chart of the HDP iteration is shown.

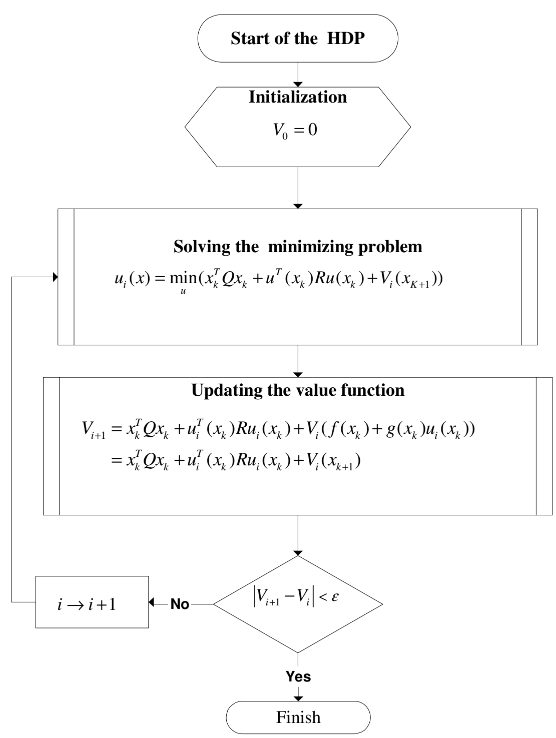

Figure 1. Flow chart shows the proposed algorithm

B. Convergence of the iteration

It has been shown that HDP iterations converge for linear systems [2][8]. In this subsection, the nonlinear case is considered as we present a proof of convergence of the iteration between (10) and (11) to $V_i \Rightarrow V^*$ and the control policy $u_i \Rightarrow u^*$ as $i \Rightarrow \infty$.

Lemma 1 Let μ_i be any arbitrary sequence of control policies, and u_i is the policies as in (10). Let V_i be as in (11) and Λ_i as

$$\Lambda_{i+1}(x_k) = x_k Q x_k + \mu_i^T R \mu_i + \Lambda_i(x_{k+1}) . \tag{12}$$

If $V_0 = \Lambda_0 = 0$, then $V_i \le \Lambda_i \quad \forall i$.

Proof : It is straight forward from the fact that V_{i+1} is a result of minimizing the right hand side of equation (10) with respect to the control input u, while Λ_i is a result of any arbitrary control input. ■

Lemma 2 Let the sequence $\{V_i\}$ be defined as in (11). If the system is controllable, then there is an upper bound Y such that $0 \le V_i \le Y \quad \forall i$.

Proof : Let $\eta(x_k)$ be any stabilizing and admissible control input, and Let $V_0 = Z_0 = 0$ where V_i is updated as (11) and Z_i is updated as

$$Z_{i+1}(x_k) = x_k Q x_k + \eta^T(x_k) R \eta(x_k) + Z_i(x_{k+1}) . \tag{13}$$

It follows that the difference

$$\begin{aligned} Z_{i+1}(x_k) - Z_i(x_k) &= Z_i(x_{k+1}) - Z_{i-1}(x_{k+1}) \\ &= Z_{i-1}(x_{k+2}) - Z_{i-2}(x_{k+2}) \\ &= Z_{i-2}(x_{k+3}) - Z_{i-3}(x_{k+3}) \\ &\;\;. \\ &\;\;. \\ &\;\;. \\ &= Z_1(x_{k+i}) - Z_0(x_{k+i}) \end{aligned} \tag{14}$$

Then (14) can be written as

$$Z_{i+1}(x_k) - Z_i(x_k) = Z_1(x_{k+i}) - Z_0(x_{k+i}),$$

Since $Z_0(x_k) = 0$, so one has

$$\begin{aligned} & Z_{i+1}(x_k) = Z_1(x_{k+i}) + Z_i(x_k) \\ & = Z_1(x_{k+i}) + Z_1(x_{k+i-1}) + Z_{i-1}(x_k) \\ & = Z_1(x_{k+i}) + Z_1(x_{k+i-1}) + Z_1(x_{k+i-1}) + Z_{i-2}(x_k) \\ & = Z_1(x_{k+i}) + Z_1(x_{k+i-1}) + Z_1(x_{k+i-2}) + ... + Z_1(x_k) \end{aligned} \tag{15}$$

so equation (15) can be written as

$$\begin{aligned} Z_{i+1}(x_k) &= \sum\nolimits_{j=0}^{i} Z_1(x_{k+j}) \\ &= \sum\nolimits_{j=0}^{i} (x_{k+j}^T Q x_{k+j} + \eta^T(x_{k+j}) R \eta(x_{k+j})) \\ &\le \sum\nolimits_{j=0}^{\infty} (x_{k+j}^T Q x_{k+j} + \eta^T(x_{k+j}) R \eta(x_{k+j})) \end{aligned} \tag{16}$$

Note that the system is stable, *i.e.* $x_k \to 0$ as $k \to \infty$, as the control input $\eta(x_k)$ is stabilizable and admissible, then

$$\forall i: \quad Z_{i+1}(x_k) \le \sum\nolimits_{i=0}^{\infty} Z_1(x_{k+i}) \le Y$$

Form Lemma 1, one has

$$\forall i: \quad V_{i+1}(x_k) \le Z_{i+1}(x_k) \le Y$$ ■

Now Lemma 1 and Lemma 2 will be used in the next main theorem.

Theorem 1 Define the sequence $\{V_i\}$ as in (11), with $V_0 = 0$. Then $\{V_i\}$ is a nondecreasing sequence in which $V_{i+1}(x_k) \ge V_i(x_k) \ \forall i$, and converge to the value function of the DT HJB, *i.e.* $V_i \Rightarrow V^*$ as $i \Rightarrow \infty$.

Proof: Let $V_0 = \Phi_0 = 0$ where V_i is updated as in (11) and, and Φ_i is updated as

$$\Phi_{i+1}(x_k) = (x_k Q x_k + u_{i+1}^T R u_{i+1} + \Phi_i(x_{k+1})) \tag{11}$$

with the policies u_i as in (10). We will first prove by induction that $\Phi_i(x_k) \le V_{i+1}(x_k)$. Note that

$$V_1(x_k) - \Phi_0(x_k) = x_k^T Q x_k \ge 0$$

$$V_1(x_k) \ge \Phi_0(x_k)$$

Assume that $V_i(x_k) \ge \Phi_{i-1}(x_k) \quad \forall x_k$. Since

$$\Phi_i(x_k) = x_k Q x_k + u_i^T R u_i + \Phi_{i-1}(x_{k+1})$$

$$V_{i+1}(x_k) = x_k Q x_k + u_i^T R u_i + V_i(x_{k+1}) ,$$

then

$$V_{i+1}(x_k) - \Phi_i(x_k) = V_i(x_{k+1}) - \Phi_{i-1}(x_{k+1}) \ge 0 ,$$

and therefore

$$\Phi_i(x_k) \le V_{i+1}(x_k) . \tag{12}$$

From Lemma 1 $V_i(x_k) \le \Phi_i(x_k)$ and therfore

$$V_i(x_k) \le \Phi_i(x_k) \le V_{i+1}(x_k)$$

$$V_i(x_k) \le V_{i+1}(x_k)$$

hence proving that $\{V_i\}$ is a nondecreasing sequence bounded from above as shown in Lemma 2. Hence $V_i \to V^*$ as $i \to \infty$. ■

We just proved that the proposed HDP algorithm converges to the value function of the DT HJB equation that appears in the nonlinear discrete-time optimal control.

C. Neural network approximation

In the case of linear systems the cost and policy are quadratic and linear respectively. In the nonlinear case, this is not necessarily true and therefore one needs to use a parametric structure or a neural network to approximate both $u_i(x)$ and $V_i(x)$. Therefore, as is standard, in order to implement the HDP iterations on equations (10) and (11) we now employ neural networks for value function approximation.

Denote the following neural networks used to approximate $V_i(x)$ and $u_i(x)$

$$\hat{V}_i(x_k, W_{Vi}) = W_{Vi}^T \phi(x_k) \tag{17}$$

$$\hat{u}_i(x_k, W_{ui}) = W_{ui}^T \sigma(x_k) \tag{18}$$

and the target cost function

$$\begin{aligned} d(\phi(x_k), W_{Vi}^T) &= x_k^T Q x_k + \hat{u}_i^T(x_k) R \hat{u}_i(x_k) + \hat{V}_i(x_{k+1}) \\ &= x_k^T Q x_k + \hat{u}_i^T(x_k) R \hat{u}_i(x_k) + W_{Vi}^T \phi(x_{k+1}) \end{aligned} \tag{19}$$

where $W_V \in \mathbb{R}^{L_v \times 1}$ and $\phi(x_k) \in \mathbb{R}^{L_v \times 1}$.

Note that in (17) the relation between the weight W_{Vi} and the target function (19) is explicit, so the parameter vector W_{Vi+1} is found by minimizing the error between the target value function (19) and (17) in a least-squares sense over a compact set Ω, and is therefore given as

$$W_{Vi+1} = \arg\min_{W_{Vi+1}} \{ \int_\Omega | W_{Vi+1}^T \phi(x_k) - d(\phi(x_k), W_{Vi}^T) |^2 \, dx_k \}. \tag{20}$$

Solving the least-squares (LS) problem one obtains

$$W_{Vi+1} = \left(\int_\Omega \phi(x_k) \phi(x_k)^T dx \right)^{-1} \int_\Omega \phi(x_k) \hat{V}_{i+1}(\phi(x_k), W_{Vi}) dx \tag{21}$$

Similarly, to find the parameters of the control policy $\hat{u}_i(x_k, W_{ui})$. They are found by solving for

$$W_{ui} = \arg\min_\alpha \left. \begin{pmatrix} x_k^T Q x_k + \hat{u}^T(x_k, \alpha) R \hat{u}(x_k, \alpha) + \\ \hat{V}_i(f(x_k) + g(x_k) \hat{u}(x_k, \alpha)) \end{pmatrix} \right|_\Omega \tag{22}$$

where $W_u \in \mathbb{R}^{L_u \times 1}$ and, $\sigma(x_k) \in \mathbb{R}^{L_u \times 1}$.

Note that the relation between the control weights W_{ui} in (22) is implicit. One can use a gradient steepest decent algorithm on a training set constructed from Ω to update the weights as

$$W_{ui(j+1)} = W_{ui(j)} - \alpha \frac{\partial (x_k^T Q x_k + \hat{u}_{i(j)}^T R \hat{u}_{i(j)} + \hat{V}_i(x_{k+1})}{\partial W_{ui(j)}} \tag{23}$$

where α is a positive stepsize. (23) can be written as

$$W_{ui(j+1)} = W_{ui(j)} -$$

$$\alpha (2\sigma(x_k) R \hat{u}_{i(j)} + \sigma(x_k) g(x_k)^T \frac{\partial \phi(x_{k+1})}{\partial x_{k+1}} W_{Vi})$$

where $x_{k+1} = f(x_k) + g(x_k) \hat{u}(x_k, W_{ui(j)})$. The weights $W_{ui(j+1)} \Rightarrow W_{ui}$ as $j \Rightarrow \infty$, which satisfies (22). Note that one can use different gradient methods like Newton's method and Levenberg-Marquardt method.

IV. Discrete-time Nonlinear System Example

In this section, two examples are provided to demonstrate the solution of the DT HJB equation. The first example will be a linear dynamical system, which is a special case of the nonlinear system. The second example is for a DT nonlinear system. MATLAB is used in the simulations to implement some of the functions discussed in the paper.

A. Linear system example

Consider the linear system

$$x_{k+1} = A x_k + B u \tag{24}$$

It is known that the solution of the optimal control problem for the linear system is quadratic in the state and given as

$$V^*(x_k) = x_k^T P x_k$$

where P is the solution of the ARE. Consider the linear system

$$A = \begin{bmatrix} 0 & .1 \\ .3 & -1 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The solution for the ARE for the given linear system is

$$P = \begin{bmatrix} 3.0714 & -0.2394 \\ -0.2394 & 3.8336 \end{bmatrix} \tag{25}$$

and the optimal control $u_k^* = L x_k$, where L is the optimal policy

$$L = [-0.2379 \quad .7981] \tag{26}$$

The control is approximated as follows

$$\hat{u}_i = W_{ui}^T \sigma(x_k) \tag{27}$$

where W_u is the weights, and the $\sigma(x_k)$ is the basis. The basis is given as

$$\sigma^T(x_k) = \begin{bmatrix} x_k(1) & x_k(2) & x_k^2(1) & 2x_k(1)x_k(2) & x_k^2(2) \end{bmatrix}$$

and the weights are

$$W_u^T = \begin{bmatrix} w_{u1} & w_{u2} & w_{u3} & w_{u4} & w_{u5} \end{bmatrix}$$

The control weights should converge to

$$\begin{bmatrix} L_{11} & L_{12} \end{bmatrix} = \begin{bmatrix} w_{u1} & w_{u2} \end{bmatrix}$$

and the other weights should be zeros

The approximation of the value function is given as

$$\hat{V}_{i+1}(x_k, W_{Vi+1}) = W_{Vi+1}^T \phi(x_k)$$

where W_V is the weight of the neural network and $\phi(x_k)$ is the neuron vector

$$\phi^T(x) = \begin{bmatrix} x_1 & x_2 & x_1^2 & 2x_1x_2 & x_2^2 \end{bmatrix}$$

and the weights are given as

$$W_V^T = \begin{bmatrix} w_{V1} & w_{V2} & w_{V3} & w_{V4} & w_{V5} \end{bmatrix}$$

In the simulation the weights of the value function are related to the P matrix given in (25) as follows

$$\begin{bmatrix} P_{11} & P_{12} \\ P_{21} & P_{22} \end{bmatrix} = \begin{bmatrix} w_{V3} & w_{V4} \\ w_{V4} & w_{V5} \end{bmatrix}$$

and $w_{V1} = 0, \quad w_{V2} = 0$.

The value function weights converge to

$$W_v = [0 \;\; 0 \;\; 3.072 \quad -0.2375 \quad 3.8442 \;].$$

The control weights converge to

$$W_u = [-0.2380 \quad 0.7983 \quad -0.0007 \quad 0.0035 \quad -0.0063]$$

Note that the value function weights converge to the solution of the ARE (25), also the control weights converge to the optimal policy (26) as expected.

B. Nonlinear system example

Consider the following affine in input nonlinear system

$$x_{k+1} = f(x_k) + g(x_k)u_k \quad (28)$$

where

$$f(x_k) = \begin{bmatrix} 0.2x_k(1)\exp(x_k^2(2)) \\ .3x_k^3(2) \end{bmatrix} \qquad g(x_k) = \begin{bmatrix} 0 \\ -.2 \end{bmatrix}$$

To approximation of the value function is given as

$$\hat{V}_{i+1}(x_k, W_{Vi+1}) = W_{Vi+1}^T \phi(x_k)$$

and the control input is approximated as

$$\hat{u}_i = W_{ui}^T \sigma(x_k)$$

The neuron vector of the Neural network that approximates the value function

$$\phi(x) = [x_1^2 \quad x_1x_2 \quad x_2^2 \quad x_1^4 \quad x_1^3x_2$$
$$x_2^4 \quad x_1^6 \quad x_1^5x_2 \quad x_1^4x_2^2 \quad x_1^3x_2^3$$
$$x_1^2x_2^4 \quad x_1x_2^5 \quad x_2^6]$$

and the weights are given as

$$W_V = [w_{V1} \quad w_{V2} \quad w_{V3} \quad w_{V4} \quad \ldots\ldots \quad w_{V15}].$$

The neuron vector of the Neural network that approximates the control is given as

$$\sigma^T(x) = [x_1 \quad x_2 \quad x_1^3 \quad x_1^2x_2 \quad x_1x_2^2$$
$$x_2^3 \quad x_1^5 \quad x_1^4x_2 \quad x_1^3x_2^2 \quad x_1^2x_2^3$$
$$x_1x_2^4 \quad x_2^5]$$

The result of the algorithm is compared to the discrete-time State Dependent Riccati Equation (SDRE) proposed in [32].

The training sets is $x_1 \in [-2,2]$, $x_2 \in [-1,1]$. The value function weights converged to the following

$$W_V^T = [1.0382 \quad 0 \quad 1.0826 \quad .0028 \quad -0 \quad -.053 \quad 0 \quad -.2792$$
$$-.0004 \quad 0 \quad -.0013 \quad 0 \quad .1549 \quad 0 \quad .3034]$$

and the control weights converged to

$$W_u^T = [\,0 \quad -.0004 \quad 0 \quad 0 \quad 0 \quad .0651 \quad 0 \quad 0 \quad 0 \quad -.0003 \quad 0 \quad -.0046]$$

In the next figures, we compare the results obtained using the SDRE and the HDP based method. Figure 2 and 3 show the states trajectories for the system for both methods.

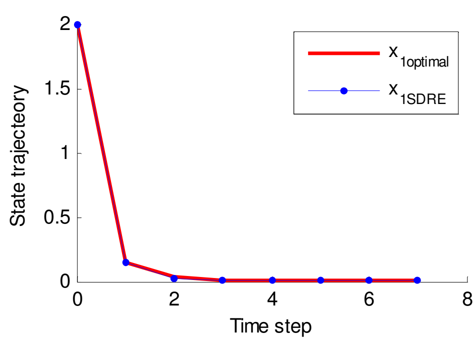

Figure 2. The state trajectory for both methods

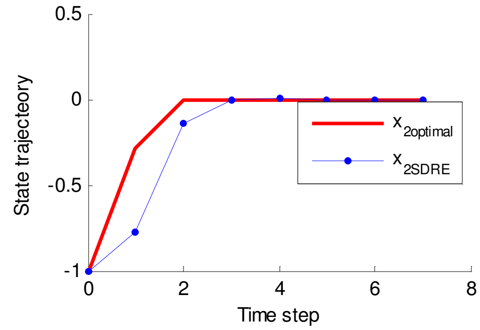

Figure 3. The state trajectory for both methods

In Figure 4, the cost function of the SDRE solution and the cost function of the proposed algorithm in this paper are compared. It is clear from the simulation that the cost function for the control policy derived from the HDP method is lower than the one obtained from the SDRE method. In figure 5, the control signal for both methods is shown.

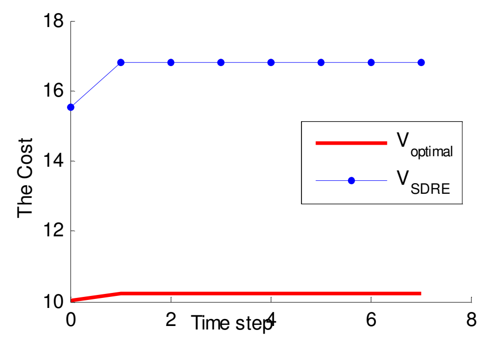

Figure 4. The state cost function for both methods

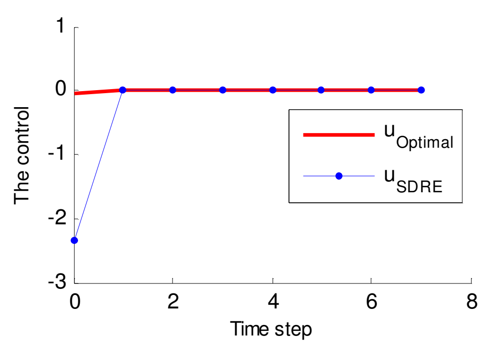

Figure 5. The control input for both methods

V. CONCLUSION

A rigorous computationally effective algorithm to find the discrete-time nonlinear optimal state feedback control laws by solving the corresponding DT HJB equation. The algorithm proposed in this paper namely Heuristic Dynamic programming (HDP) is used to find the optimal controller.

The main contribution in this paper is the proof of convergence for the HDP algorithm to the value function of DT HJB.

Neural networks are used as parametric structures to approximate the critics, *i.e.* $\hat{V}_i$, and the actors networks, *i.e.* $\hat{u}_i$. In the simulation part it is shown that the linear system critic network converges to the solution of the ARE, and the actor network converges to the optimal policy. In the nonlinear example, it is shown that the optimal controller derived from the HDP based method outperforms that derived using the discrete-time SDRE method.

REFERENCES

[1] S. J. Bradtke, B. E. Ydestie, A. G. Barto, "Adaptive linear quadratic control using policy iteration," Proceedings of the American Control Conference , pp. 3475-3476, Baltmore, Myrland, June, 1994.

[2] A. Al-Tamimi, M. Abu-Khalaf, F. L. Lewis, "Adaptive Critic Designs for Discrete-Time Zero-Sum Games with Application to H-Infinity Control" *IEEE Transactions on Systems, Man, Cybernetics-Part B, November* 2006.

[3] A. Al-Tamimi, M. Abu-Khalaf, F. L. Lewis, "Model-Free Q-Learning Designs for Discrete-Time Zero-Sum Games with Application to H-Infinity Control," *to appear, Automatica.*

[4] J. Si, A. Barto, W. Powell, D. Wunsch, *Handbook of Learning and Approximate Dynamic Programming*, John Wiley, New Jersey, 2004.

[5] S. Hagen, B Krose, " Linear quadratic Regulation using Reinforcement Learning," Belgian_Dutch Conference on Mechanical Learning, pp. 39-46, 1998.

[6] D. Kleinman, " Stabilizing a discrete, Constant, Linear System with Application to iterative Methods for Solving the Riccati Equation," IEEE Trans. Automat. Control, pp. 252-254, June 1974.

[7] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike elements that can solve difficult learning control problems," *IEEE Trans. Syst., Man, Cybern.,* vol. SMC-13, pp. 835–846, 1983.

[8] T. Landelius, *Reinforcement Learning and Distributed Local Model Synthesis,* PhD Dissertation, Linkoping University, Sweden, 1997.

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

[10] C. Watkins, *Learning from Delayed Rewards*, Ph.D. Thesis, Cambridge University, Cambridge, England, 1989.

[11] P.J. Werbos, "Neural networks for control and system identification," Heuristics, Vol. 3, No. 1, Spring 1990, pp. 18-27,.1990.

[12] P.J. Werbos., "A menu of designs for reinforcement learning over time," , Neural Networks for Control, pp. 67-95, ed. W.T. Miller, R.S. Sutton, P.J. Werbos, Cambridge: MIT Press, 1991.

[13] R. Howard, *Dynamic Programming and Markov Processes*, MIT Press, Cambridge, MA, 1960.

[14] 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.

[15] W. Lin and C. I. Byrnes, "H_∞ Control of Discrete-Time Nonlinear System,"IEEE Trans. on Automat. Control , vol 41, No 4, pp 494-510, 1996. 1994.

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

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

[18] K.S. Narendra and F.L. Lewis,"Special Issue on Neural Network feedback Control," Automatica, vol. 37, no. 8, Aug. 2001.

[19] M. Abu-Khalaf, F. L. Lewis, and J. Huang, "Hamilton-Jacobi-Isaacs formulation for constrained input nonlinear systems," in 43rd IEEE Conference on Decision and Control, 2004, pp. 5034 - 5040 Vol.5, Bahamas, 2004.

[20] 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, pp. 779 – 791, 2005.

[21] B. Widrow, N. Gupta, and S. Maitra, "Punish/reward: Learning with a critic in adaptive threshold systems," IEEE Trans. Syst., Man, Cybern., vol. SMC-3, pp. 455–465, 1973.

[22] W. H Kwon and S. Han, *Receding Horizon Control*, Springer-Verlag, London, 2005.

[23] B. Stevens, F. L. Lewis, *Aircraft Control and Simulation*, 2nd edition, John Wiley, New Jersey, 2003.

[24] F.L. Lewis, *Optimal Estimation,* John Wiley, New York, 1986.

[25] F. L. Lewis, *Applied Optimal Control and Estimation,* Prentice-Hall, New Jersey, 1992.

[26] J. J. Murray, C. J. Cox, G. G. Lendaris, and R. Saeks, "Adaptive Dynamic Programming," IEEE Trans. on Sys., Man. and Cyb., Vol. 32, No. 2, pp 140-153, 2002.

[27] J. Si and Wang, "On-Line learning by association andreinforcement, IEEE Trans. Neural Networks, vol. 12, pp.264-276, Mar. 2001

[28] Xi-Ren Cao, "Learning and Optimization—From a SystemsTheoretic Perspective", Proc. of IEEE Conference on Decision and Control, pp. 3367-3371, 2002

[29] P. He and S. Jagannathan, "Reinforcement learning-basedoutput feedback control of nonlinear systems with input constraints," IEEE Trans. Systems, Man, and Cybernetics -Part B:Cybernetics, vol. 35, no.1, pp. 150-154, Feb. 2005

[30] Chen, Z., Jagannathan, S,," Neural Network -based Nearly Optimal Hamilton-Jacobi-Bellman Solution for Affine Nonlinear Discrete-Time Systems," IEEE CDC 05 ,pp 4123-4128, Dec 2005

[31] J. Huang," An algorithm to solve the discrete HJI equation arising in the L_2 gain optimization problem," INT. J. Control, Vol 72, No 1, pp 49-57, 1999

[32] Cloutier, J. R.," State –Dependent Riccati equation Techniques: An overview," Proceeding of the American control conference, Albuquerque, NM, June 4-6, 1997, pp 932-936.