Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

Continuous-Time Single Network Adaptive Critic for Regulator Design of Nonlinear Control Affine Systems ⋆

Swagat Kumar * **Radhakant Padhi** ** **Laxmidhar Behera** *

* *Department of Electrical Engineering, Indian Institute of Technology Kanpur, Uttar Pradesh, India - 208 016. e-mail: {swagatk, lbehera}@iitk.ac.in, l.behera@ulster.ac.uk*
** *Department of Aerospace Engineering, Indian Institute of Science, Bangalore, Karnataka, India - 560 012. e-mail: padhi@aero.iisc.ernet.in*

Abstract: An optimal control law for a general nonlinear system can be obtained by solving Hamilton-Jacobi-Bellman equation. However, it is difficult to obtain an analytical solution of this equation even for a moderately complex system. In this paper, we propose a continuous-time single network adaptive critic scheme for nonlinear control affine systems where the optimal cost-to-go function is approximated using a parametric positive semi-definite function. Unlike earlier approaches, a continuous-time weight update law is derived from the HJB equation. The stability of the system is analysed during the evolution of weights using Lyapunov theory. The effectiveness of the scheme is demonstrated through simulation examples.

Keywords: Adaptive optimal control, HJB, single network adaptive critic, control-affine systems.

1. INTRODUCTION

In case of nonlinear systems, one of the main focus of the control design processes available in literature is to ensure stability of the system while achieving good trajectory tracking accuracy. Many times however, simple stability of the system is not good enough and optimality issues should be addressed at so as not to end up with an impracticable control design. This gives rise to optimal control methodologies where one tries to design controllers that minimize certain meaningful performance indices. While the optimal control theory is quite well-established, its application to control of nonlinear systems has been limited owing to the mathematical complexity involved in finding closed form solutions to the control variable in state feedback form. Bellman's dynamic programming [Naidu, 2003, Bryson and Ho, 1975] treats such optimal control problems as multistage decision making processes, where a decision is chosen from a finite number of decisions. The continuous-time analog of Bellman's recurrence equation in dynamic programming is called the *Hamilton-Jacobi-Bellman Equation*. This equation, in general, is a nonlinear partial differential equation which is difficult to solve.

In discrete-time, dynamic programming problem is solved backwards in time. Quite recently, a number of architectures have been reported in literature, collectively known as 'Adaptive Critic' which solves this dynamic programming problem in forward direction of time. It is also known as forward dynamic programming or *Approximate dynamic programming* [Si et al., 2005, Ch. 3]. Adaptive critic based methods have two components - an actor which computes the control action and a critic which evaluates its performance. Based on the feedback received from the critic, the actor improves its performance in the next step. Various architectures as well as learning algorithms for actor and critic have been proposed in last few years. An interested reader may refer to [Prokhorov and II, 1997] and [Si et al., 2005] for details. Quite recently, Padhi et. al. [Padhi et al., 2006] introduced a simplified version of adaptive critic architecture which uses only one network instead of two required in a standard adaptive critic design. This architecture is called "single network adaptive critic (SNAC)". This architecture can be applied to a class of systems where control input can be expressed explicitly in terms of state and costate variables.

In this paper, we introduce a variant of continuous-time adaptive critic structure for controlling nonlinear affine systems. It is well known that the HJB equation is necessary as well as sufficient condition for optimality [Bryson and Ho, 1975, Naidu, 2003]. However, finding an analytical solution of HJB equation is usually very difficult even for a moderately complex system.

We approximate this optimal cost function using a parametric positive semi-definite function. This parametric

⋆ This work was supported by Department of Science and Technology (DST), Govt. Of India under the project titled "Intelligent Control Schemes and application to dynamics and visual control of redundant manipulator systems". The project number is DST/EE/20050331. Dr. Laxmidhar Behera is an associate professor at Department of Electrical Engineering, IIT Kanpur. Currently, he is a reader at School of computing and intelligent systems, University of Ulster, UK.

function may also be replaced with a suitable neural network. Now, a continuous-time weight update law is derived so as to satisfy the HJB equation. This gives rise to an under-determined linear least square problem which can be solved accurately using standard numerical routines. It is also shown that the system is stable in the sense of Lyapunov during evolution of weights. The training is carried out in an online fashion where weights attain their final value during the closed loop operation of the system itself. In that respect, the critic does not require any separate training phase. The performance of proposed algorithm is analyzed for both linear and nonlinear affine systems and various related issues are discussed. In case of linear systems, it is shown that the solution converges to that of Algebraic Riccati Equation (ARE), provided the system parameters are initialized properly. In case of nonlinear systems, linear optimal controllers are derived and their performance is compared with those of LQR controllers for their linearized models. The local optimality is verified through simulations.

The paper is organized as follows. The proposed scheme is presented in Section 2 followed by its stability analysis in Section 3. The simulation results are provided in Section 4 and appropriate conclusions are drawn in Section 5.

2. CONTINUOUS-TIME SINGLE NETWORK ADAPTIVE CRITIC SCHEME

Consider a nonlinear control-affine system given by

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

The task is to find a control input that minimises the performance index given by

$$J(\mathbf{x}(t_0), t_0) = S(\mathbf{x}(t_f), t_f) + \int_{t_0}^{t_f} \psi[\mathbf{x}(\tau), \mathbf{u}(\tau)]d\tau \tag{2}$$

along with the boundary conditions

$$\mathbf{x}(t_0) = \mathbf{x}_0 \text{ is fixed and } \mathbf{x}(t_f) \text{ is free.} \tag{3}$$

and the utility function ψ is given by

$$\psi(\mathbf{x}, \mathbf{u}) \triangleq \frac{1}{2}[\mathbf{x}^T Q\mathbf{x} + \mathbf{u}^T R\mathbf{u}] \tag{4}$$

Let us define a scalar function $J^*(\mathbf{x}^*(t), t)$ as the *optimal* value of the performance index J for an initial state $\mathbf{x}^*(t)$ at time t, i.e.,

$$J^*(\mathbf{x}^*(t), t) = S(\mathbf{x}(t_f), t_f) + \int_t^{t_f} \psi(\mathbf{x}^*(\tau), \mathbf{u}^*(\tau), \tau)d\tau \tag{5}$$

Consider a Hamiltonian given by

$$H(\mathbf{x}, \boldsymbol{\lambda}^*, \mathbf{u}) = \psi(\mathbf{x}, \mathbf{u}) + \boldsymbol{\lambda}^{*T}[\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}] \tag{6}$$

where $\boldsymbol{\lambda}^* = \frac{\partial J}{\partial \mathbf{x}}^*$. The optimal control is obtained from the necessary condition given by

$$\frac{\partial H}{\partial \mathbf{u}} = \frac{\partial \psi}{\partial \mathbf{u}} + \boldsymbol{\lambda}^{*T}\frac{\partial}{\partial \mathbf{u}}[\mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}] = 0 \tag{7}$$

This gives the following optimal control equation for control affine system described in (1):

$$\mathbf{u} = -R^{-1}\mathbf{g}^T\boldsymbol{\lambda}^* \tag{8}$$

Substituting the value of $\mathbf{u}$ into (6), we get

$$H(\mathbf{x}^*, \boldsymbol{\lambda}^*, \mathbf{u}^*) = \frac{1}{2}\mathbf{x}^{*T}Q\mathbf{x}^* + \frac{1}{2}\boldsymbol{\lambda}^{*T}\mathbf{g}R^{-1}\mathbf{g}^T\boldsymbol{\lambda} + \boldsymbol{\lambda}^{*T}[\mathbf{f} - \mathbf{g}R^{-1}\mathbf{g}^T\boldsymbol{\lambda}^*] \tag{9}$$

On simplification, we have following optimal Hamiltonian:

$$\begin{aligned} H^* &= \frac{1}{2}\mathbf{x}^{*T}Q\mathbf{x}^* - \frac{1}{2}\boldsymbol{\lambda}^{*T}\mathbf{g}R^{-1}\mathbf{g}^T\boldsymbol{\lambda}^* + \boldsymbol{\lambda}^{*T}\mathbf{f} \\ &= \frac{1}{2}\mathbf{x}^{*T}Q\mathbf{x}^* - \frac{1}{2}\boldsymbol{\lambda}^{*T}G\boldsymbol{\lambda} + \boldsymbol{\lambda}^{*T}\mathbf{f} \end{aligned} \tag{10}$$

where $G = \mathbf{g}R^{-1}\mathbf{g}^T$. We know that the optimal value function $J^*(\mathbf{x}^*, t)$ must satisfy the *Hamilton-Jacobi-Bellman* (HJB) equation given by

$$\frac{\partial J^*}{\partial t} + \min_{\mathbf{u}} H(\mathbf{x}, \frac{\partial J^*}{\partial \mathbf{x}}, \mathbf{u}, t) = 0 \tag{11}$$

with boundary condition given by

$$J^*(\mathbf{x}^*(t_f), t_f) = S(\mathbf{x}^*(t_f), t_f) \tag{12}$$

It provides the solution to the optimal control problem for general nonlinear dynamical systems. However, the analytical solution to the HJB equation is difficult to obtain in most cases. It is well known that the HJB equation is both necessary as well as sufficient condition of optimality [Naidu, 2003, ch. 2, pp. 286-287]. Therefore by combining (10) and (11) we can say that, in case of control affine systems (1), the optimal value function must satisfy following nonlinear dynamic equation:

$$\frac{\partial J^*}{\partial t} + \frac{1}{2}\mathbf{x}^{*T}Q\mathbf{x}^* - \frac{1}{2}\left(\frac{\partial J^*}{\partial \mathbf{x}}\right)^T G\frac{\partial J^*}{\partial \mathbf{x}} + \left(\frac{\partial J^*}{\partial \mathbf{x}}\right)^T \mathbf{f} = 0 \tag{13}$$

Since, the analytical solution of the above equation is difficult, we take a different approach and approximate the optimal value function as follows:

$$V(\mathbf{x}, t) = h(\mathbf{w}, \mathbf{x}) \tag{14}$$

where the approximating function $h(\mathbf{w}, \mathbf{x})$ is selected so as to satisfy certain initial conditions stated in next section. The parameter t has been put in $V(\mathbf{x}, t)$ to show explicit dependence of value function on time because of time varying parameters $\mathbf{w}$ in the approximating function $h(\mathbf{w}, \mathbf{x})$.

For the value function given in (14) to be optimal, it must satisfy the HJB equation (13). This gives

$$\frac{\partial V}{\partial t} + \psi(\mathbf{x}, \mathbf{u}) + \left(\frac{\partial V}{\partial \mathbf{x}}\right)^T [\mathbf{f} + \mathbf{g}\mathbf{u}] = 0 \tag{15}$$

$$\frac{\partial h}{\partial \mathbf{w}}\dot{\mathbf{w}} + \frac{1}{2}\mathbf{x}^T Q\mathbf{x} - \frac{1}{2}\left(\frac{\partial V}{\partial \mathbf{x}}\right)^T G\frac{\partial V}{\partial \mathbf{x}} + \left(\frac{\partial V}{\partial \mathbf{x}}\right)^T \mathbf{f} = 0 \tag{16}$$

This gives following weight update law:

$$\frac{\partial h}{\partial \mathbf{w}}\dot{\mathbf{w}} = -\frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \frac{1}{2}\frac{\partial h}{\partial \mathbf{x}}^T G\frac{\partial h}{\partial \mathbf{x}} - \left(\frac{\partial h}{\partial \mathbf{x}}\right)^T \mathbf{f} \tag{17}$$

The task is to find $\dot{\mathbf{w}}$ so that the above scalar equation is satisfied. This is an *under-determined* system of linear equations with number of equations less than the number of variables to be estimated. Though, there are infinitely many solutions for $\dot{\mathbf{w}}$ which would exactly satisfy the above equation, we seek the one which minimises $\|\dot{\mathbf{w}}\|_2$. The problem is referred to as finding *minimum norm* solution to an *under-determined* system of linear equations. Pseudo-inverse method is used to solve this problem.

Equation (17) may be written as

$$\mathbf{s}\dot{\mathbf{w}} = r \tag{18}$$

where $\mathbf{s} = \frac{\partial h}{\partial \mathbf{w}}$ is a $1 \times N_w$ a vector and $r = -\frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \frac{1}{2}\frac{\partial h}{\partial \mathbf{x}}^T G\frac{\partial h}{\partial \mathbf{x}} - \left(\frac{\partial h}{\partial \mathbf{x}}\right)^T \mathbf{f}$ is a scalar quantity. The pseudoinverse solution is given by

$$\dot{\mathbf{w}} = \mathbf{s}^T(\mathbf{s}\mathbf{s}^T)^{-1}r \tag{19}$$

Note that the term $\mathbf{s}\mathbf{s}^T$ is a scalar quantity and its inverse is easily computable. The control scheme is shown in Figure 1. The blocks are self-explanatory.

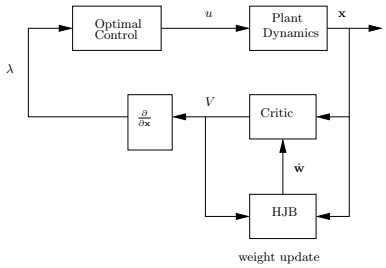


Fig. 1. Continuous-time single network adaptive critic scheme

3. STABILITY ANALYSIS

The link between stability and optimality is well known. The value function for a meaningful optimal stabilization problem is also a Lyapunov function for the closed-loop system. In other words, *every meaningful value function is a Lyapunov function* [Freeman and Kokotovic, 1996]. In the previous section, we saw that the optimal value function is approximated using a parametric function $h(\mathbf{w}, \mathbf{x})$. The parametric function is selected so as to satisfy following initial conditions:

$$V(0,t) = h(0,\mathbf{w}) \geq 0 \;\; \forall t \;\geq 0 \tag{20a}$$

$$\frac{\partial V}{\partial \mathbf{x}}(\mathbf{x},t) = \frac{\partial h}{\partial \mathbf{x}} = 0, \quad \text{when } \mathbf{x} = 0 \tag{20b}$$

The condition (20a) may be replaced by the condition that the function $V(\mathbf{x}, t)$ be lower bounded. Note that the optimal control is a function of $\frac{\partial V}{\partial \mathbf{x}}$ as shown in equation (8) and the condition (20b) is needed to ensure that the control input becomes zero only when state $\mathbf{x}$ approaches zero value.

In order to analyze the stability of the scheme, we consider (14) as a Lyapunov function candidate which satisfies the conditions given by (20). Because of time-varying weight parameters, we have a non-autonomous system and thus the Lyapunov function candidate is considered to have explicit time-dependence.

The asymptotic stability analysis of non-autonomous systems is generally much harder than that of autonomous systems. In order to analyze the stability of the scheme, we make use of *Barbalat's* Lyapunov-like Lemma [Slotine and Li, 1991] which tells that if a scalar function $V(\mathbf{x}, t)$ satisfies the following conditions:

- $V(\mathbf{x}, t)$ is lower bounded
- $\dot{V}(\mathbf{x}, t)$ is negative semi-definite
- $\dot{V}(\mathbf{x}, t)$ is uniformly continuous in time

then $\dot{V}(\mathbf{x}, t) \to 0$ as $t \to \infty$.

Since the approximating function $h(\mathbf{w}, \mathbf{x})$ is chosen so as to satisfy the condition (20a), the first requirement of the above lemma is already met by choice. Differentiating $V(\mathbf{x}, t)$ with respect to time, we get

$$\dot{V} = \frac{\partial V}{\partial t} + \left(\frac{\partial V}{\partial \mathbf{x}}\right)^T \dot{\mathbf{x}} = \frac{\partial V}{\partial t} + \left(\frac{\partial V}{\partial \mathbf{x}}\right)^T [\mathbf{f} + \mathbf{g}\mathbf{u}] \tag{21}$$

Using (15) and (21), we get

$$\dot{V} = -\psi(\mathbf{x}, \mathbf{u}) = -\frac{1}{2}\mathbf{x}^T Q \mathbf{x} - \frac{1}{2}\frac{\partial V}{\partial \mathbf{x}}^T G \frac{\partial V}{\partial \mathbf{x}} \tag{22}$$

where $G = \mathbf{g}R^{-1}\mathbf{g}^T$ is a positive semi-definite matrix and $V(\mathbf{x}, t)$ is a function of both $\mathbf{x}$ and $\mathbf{w}$. $\dot{V} = 0$ when either $\{\mathbf{x} = 0, \;\; \mathbf{w} = 0\}$ or $\{\mathbf{x} = 0, \;\; \mathbf{w} \neq 0\}$ and $\dot{V} < 0$ whenever $\mathbf{x} \neq 0$. Thus, $\dot{V}$ is only negative semi-definite. Differentiating (22) once again with respect to time, we get

$$\ddot{V} = -\mathbf{x}^T Q \dot{\mathbf{x}} - \frac{\partial V}{\partial \mathbf{x}}^T G \frac{\partial^2 V}{\partial t \partial \mathbf{x}} - \frac{1}{2}\frac{\partial V}{\partial \mathbf{x}}^T \frac{\partial G}{\partial t}\frac{\partial V}{\partial \mathbf{x}} \tag{23}$$

By Lyapunov stability theory we know that the negative semi-definiteness of $\dot{V}$ ensures boundedness of $\mathbf{x}$ as well as $\dot{\mathbf{x}}$. The partial derivative $\frac{\partial V}{\partial \mathbf{x}}$ is a function of $\mathbf{w}$ and $\mathbf{x}$. $\mathbf{w}(t)$ is bounded as long as $\mathbf{x}$ is bounded and the norm $\|\frac{\partial h}{\partial \mathbf{w}}\|$ in equation (17) is non-zero and finite. The boundedness of $\mathbf{w}$ and $\mathbf{x}$ is guaranteed as long as the first two conditions of Barbalat's Lemma are met. Since $\mathbf{g}$ is assumed to be a continuous function of x as well as t, it is bounded as long as x is bounded. Thus, $\frac{\partial G}{\partial t} = 2\mathbf{g}R^{-1}\dot{\mathbf{g}}$ is also a continuous and bounded function. Thus, it can always be ensured that $\ddot{V}$ is always bounded and finite, at least for quadratic value functions. Now, by invoking Barbalat's Lemma, we find that $\dot{V} \to 0$ as $t \to \infty$. This gives,

$$\dot{V} = 0 \Rightarrow \frac{1}{2}\mathbf{x}^T Q \mathbf{x} + \frac{1}{2}\frac{\partial V}{\partial \mathbf{x}}^T G \frac{\partial V}{\partial \mathbf{x}} = 0$$

Since, both terms in the later equation are positive scalars, the above equation leads to

$$\mathbf{x}^T Q \mathbf{x} = 0 \text{ and } \frac{\partial V}{\partial \mathbf{x}}^T G \frac{\partial V}{\partial \mathbf{x}} = 0$$

Thus, we can conclude that $\mathbf{x} \to 0$ and $\frac{\partial V}{\partial \mathbf{x}} \to 0$ as $t \to \infty$.

This establishes the fact that the approximate value function (14) is a Lyapunov function and the weight update law (17) ensures asymptotic stability ($\mathbf{x} = 0$).

3.1 Discussion

Since the HJB equation (11) along with boundary condition (12) can be solved by backward integration, the weight vector $\mathbf{w}$ is updated as follows:

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \dot{\mathbf{w}}dt \tag{24}$$

where $\dot{\mathbf{w}}$ is obtained by solving the under-determined equation (17). It is also possible to integrate the differential equation (17) by Fourth-order Runge-Kutta method for better accuracy. The negative sign shows a backward integration in time. It is to be noted that, even though above update law represents a back integration process, it can still be implemented in forward time. The steps involved are enumerated below:

(1) Values for initial states are selected from the domain of interest. The weight parameters of value function are initialized so that the initial control action stabilizes the closed loop system.
(2) The control action is computed using equation (8). The system response is obtained by integrating the dynamic equation (1). Using Euler integration, we can write the state evolution as

$$\mathbf{x}(t+1) = \mathbf{x}(t) + \dot{\mathbf{x}}\, dt \tag{25}$$

(3) The under-determined equation (17) is solved using pseudo-inverse method and $\dot{\mathbf{w}}$ is given by (19). Now, the weights are updated using equation (24).
(4) The time quantity is incremented as $t = t+dt$ and the above two steps are repeated until the weights attain their steady state value. For time-invariant systems, weights should attain constant values.

As one can see, even though the system evolves forward in time, the weights are updated backwards in time. The entire training process can be carried out in real-time with a weight update law given by (24). The nature of weight update law is such that it solves the HJB equation.

4. SIMULATION AND RESULTS

In this section, we solve optimal control problem for two control affine systems - a linear and a nonlinear system. In Linear system case, we show that a quadratic value function structure gives rise to LQR control using this method. However in case of nonlinear systems, the optimal control depends on the structure of the approximating function. For a quadratic structure for the value function, we can only get a linear PD type controller. This can be seen from optimal control equation (8) which depends on $\frac{\partial V}{\partial \mathbf{x}}$. For a quadratic value function, its derivative would be a linear function of states. Hence in the following examples, we aim to search for optimal PD controllers corresponding to the structure of value function selected for the problem. Through simulation, it is shown that the performance of proposed controllers are not different from those of LQR control action derived from their linearized models.

4.1 Linear Systems

Consider a single input linear system of the form $\dot{\mathbf{x}} = A\mathbf{x} + \mathbf{b}u$ given by

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0.4 & 0.1\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix} u \tag{26}$$

The task is to find a control law $u = c(\mathbf{x})$ that minimizes the cost function

$$J = \frac{1}{2}\int_0^\infty [\mathbf{x}^T Q\mathbf{x} + u^T Ru]\, dt \tag{27}$$

where

$$Q = \begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix} \qquad \text{and} \quad R = 1$$

We know that the optimal value function for a linear system is given by

$$V = \frac{1}{2}\mathbf{x}^T P\mathbf{x} \tag{28}$$

where P is a symmetric positive definite matrix. It is trivial to show that the HJB equation (11) for this value function gives rise to Differential Riccati Equation (DRE), given by

$$\dot{P} = -(PA + A^T P) - Q + P^T BR^{-1}B^T P \tag{29}$$

and for infinite time, $\dot{P} = 0$ and above equation gives rise to Algebraic Riccati Equation (ARE). In order to solve this problem using proposed approach, we rewrite the optimal value function as

$$V = \frac{1}{2}(w_1 x_1^2 + w_2 x_2^2 + 2w_3 x_1 x_2) \tag{30}$$

where the initial value of weight vector $\mathbf{w} = [w_1 \;\; w_2 \;\; w_3]^T$ is chosen so that V is at least positive semi-definite in the beginning. The derivative of the weight vector $\dot{\mathbf{w}}$ is obtained by solving the under-determined equation (17) which is reproduced here for convenience

$$\frac{\partial V}{\partial \mathbf{w}}\dot{\mathbf{w}} = -\frac{1}{2}\mathbf{x}^T Q\mathbf{x} + \frac{1}{2}\frac{\partial V}{\partial \mathbf{x}}^T \bar{B}\frac{\partial V}{\partial \mathbf{x}} - \frac{\partial V}{\partial \mathbf{x}}A\mathbf{x} \tag{31}$$

where $\bar{B} = \mathbf{b}R^{-1}\mathbf{b}^T$ and the partial derivatives are given as follows:

$$\frac{\partial V}{\partial \mathbf{x}}^T = [w_1 x_1 + w_3 x_2 \;\; w_2 x_2 + w_3 x_1] \tag{32}$$

$$\frac{\partial V}{\partial \mathbf{w}}^T = \begin{bmatrix}0.5x_1^2 & 0.5x_2^2 & x_1 x_2\end{bmatrix} \tag{33}$$

The control law is given by (8) and for this problem, it is computed to be

$$u = -R^{-1}\mathbf{b}^T\frac{\partial V}{\partial \mathbf{x}} = -(w_2 x_2 + w_3 x_1) \tag{34}$$

The weights are updated by (24). The final values of weights after training is given below:

$$\mathbf{w} = [2.10456 \;\; 2.09112 \;\; 1.4722]^T$$

The equation (30) may be written as

$$V = \frac{1}{2}\mathbf{x}^T W\mathbf{x} = \frac{1}{2}\mathbf{x}^T\begin{bmatrix}w_1 & w_3\\ w_3 & w_2\end{bmatrix}\mathbf{x} \tag{35}$$

It can be verified that the matrix W is same as the Riccati matrix P obtained by solving the ARE as shown below.

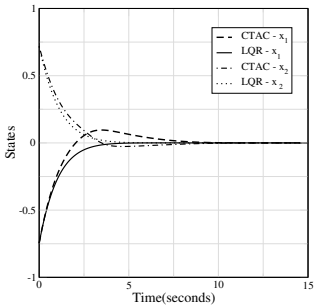
$$P = \begin{bmatrix}2.10456 & 1.4722\\ 1.4722 & 2.09112\end{bmatrix}$$

Discussion:

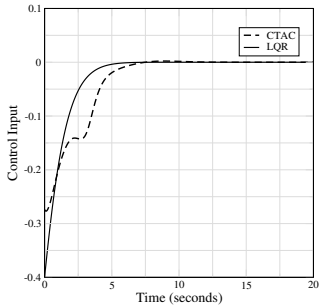
- Through this example, we show an alternative method to solve differential Riccati equation and in case of linear time-invariant systems, it is possible to obtain optimal control through this scheme.
- Note that the convergence of the weight update law (31) to Riccati solution depends on proper initialization of weights and states. Some additional constraint might be imposed on the weight values so that the current method always yields Riccati solution.
- The phase during which weights evolve, we call it a training phase. Testing phase is the one where weights have settled down to some steady state value. Evolution of states, weights as well as control during training phase is shown in Figure 2. In Figures 2(a), 2(b) and 2(d), the performance is compared with those of LQR controller. The objective is to show that the performance of the proposed control scheme do not differ too much from LQR performance during closed loop operation. Once weights attain their final value, the performance exactly matches with that of LQR control scheme. The evolution of weights during training is shown in Figure 2(c). The weight update law is given by (24) where $\dot{\mathbf{w}}$ is obtained by solving equation (31). It is to be noted that weights also evolve in the forward direction as states do, however in the process of evolution, it tends to solve HJB equation in the backward direction.

4.2 Non-linear System

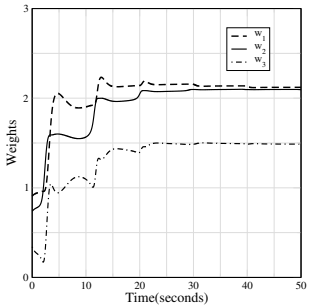
Nonlinear System Consider the following Single Link manipulator system given by



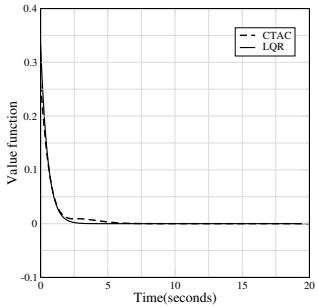
(a) Evolution of States



(b) Control Input



(c) Evolution of weights



(d) Value function

Fig. 2. Linear System: Comparison with LQR performance during training

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -10 \sin x_1 + u \tag{36}$$

We seek to find a controller that minimizes following cost function:

$$J = \frac{1}{2}\int_0^\infty [\mathbf{x}^T Q \mathbf{x} + u^T R u] dt \tag{37}$$

where

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \quad R = 1$$

We consider following structure for the optimal cost-to-go function:

$$V = \frac{1}{2}(w_1 x_1 + w_2 x_2)^2 + \frac{1}{2}(w_1^2 + w_2^2) \tag{38}$$

The corresponding derivative terms are given by

$$\frac{dV}{d\mathbf{w}} = [(w_1 x_1 + w_2 x_2)x_1 + w_1 \quad (w_1 x_1 + w_2 x_2)x_2 + w_2]^T$$
$$\frac{dV}{d\mathbf{x}} = [(w_1 x_1 + w_2 x_2)w_1 \quad (w_1 x_1 + w_2 x_2)w_2]^T \tag{39}$$

Considering the cost-to-go function (38) as a Lyapunov candidate and equating its time-derivative to the utility function, we get following under-determined equation for $\dot{\mathbf{w}}$:

$$\dot{V} = \frac{\partial V}{\partial \mathbf{w}}\dot{\mathbf{w}} + \frac{\partial V}{\partial \mathbf{x}}\dot{\mathbf{x}} = -\frac{1}{2}[\mathbf{x}^T Q \mathbf{x} + u^T R u]$$
$$\frac{\partial V}{\partial \mathbf{w}}\dot{\mathbf{w}} = -\frac{1}{2}\mathbf{x}^T Q \mathbf{x} - \frac{1}{2}u^T R u - \frac{\partial V}{\partial \mathbf{x}}\dot{\mathbf{x}} \tag{40}$$

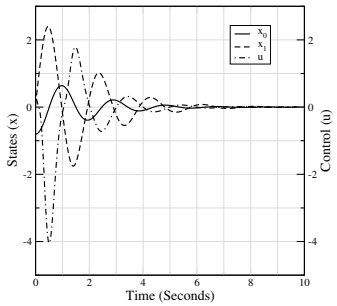
The control input is given by (8) and is computed to be:

$$u = -R^{-1}\mathbf{g}^T \frac{\partial V}{\partial \mathbf{x}} = -(w_1 x_1 + w_2 x_2)w_2 \tag{41}$$

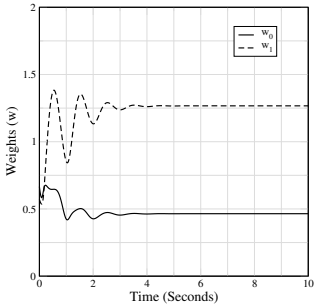
The corresponding the system response during training as well as testing phases are shown in Figures 3 and 4 respectively.

Discussion:

- Training is carried out as per steps enumerated in Section 3.1 and final values of weights are used to control the plant. Figure 3 shows the evolution of states as well as weights during training. It is to be noted that the training is not carried out for all initial conditions in a domain of interest. The training is carried out only for a single set of initial conditions of states and weights until weights settle down to their steady state values as shown in Figure 3(b). The initial values of weights must be chosen so as to render the system stable at the start of training phase.
- Figure 4 shows the system behaviour during testing phase where the weights have already attained their steady-state value. Here, its performance is compared with that of LQR control action and its seen that the performances are quite similar to each other. Note that we are using LQR control action for the nonlinear plant and the comparison is provided to show that the proposed control's behaviour is not different from that of LQR control action.
- In order to judge the local optimality of the controller, we perturb the final weights by ± 0.5 and compute the total cost over a time-interval of 20 seconds. For two weights, nine (3×3) such combinations are possible. The corresponding cost curves are plotted in Figure 5. The curve for unperturbed weights is represented by the label 'C' while the cost for LQR control is labelled as 'C_{LQR}'. The curves with perturbed weights are labelled as $C_1, \ldots, C_9$. As can be seen, the original weights incur minimum cost among all other combinations. This is of course higher than that of cost for LQR control.

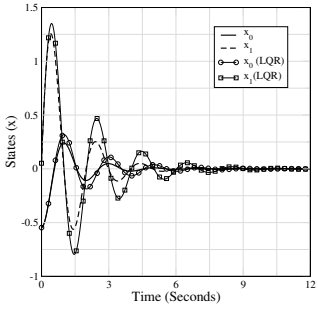


(a) Evolution of states

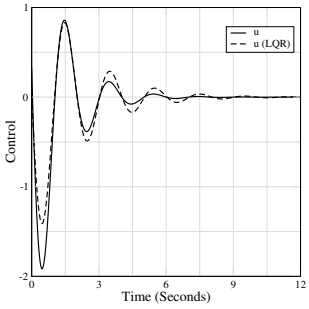


(b) Evolution of weights

Fig. 3. Nonlinear System 2: Training phase



(a) Evolution of states



(b) Control Input

Fig. 4. Nonlinear System 2: Testing phase

- Since the choice of optimal cost function is a quadratic one, we get a linear (PD) control action for the system. Figure 5 at least establishes local optimality for the given controller. The controller is optimal with respect to the structure of optimal value function chosen.

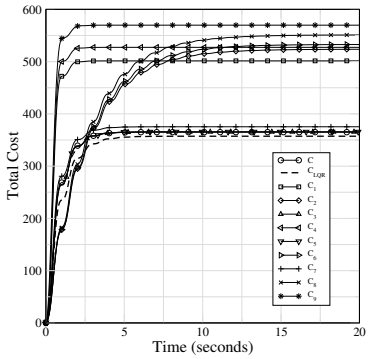


Fig. 5. Cost comparison for perturbed weights

5. CONCLUSION

In this paper, a new approach to single network adaptive critic (SNAC) is presented where optimal cost-to-go function is approximated using a quadratic polynomial function of states as well as weights. Unlike earlier approaches, a continuous-time weight update law is derived using HJB equation and stability is analyzed during evolution of weights. The training is carried out in an online fashion where states and weights evolve forward in time. The controller attains its optimal value as training proceeds. The performance of the proposed scheme is analyzed through simulations on second order linear and nonlinear control affine systems. The local optimality of the controller is verified through simulation plots.

REFERENCES

A. E. Bryson and Y. C. Ho. *Applied Optimal Control*. Taylor and Francis, 1975.

R. A. Freeman and P. V. Kokotovic. Inverse optimality in robust stabilization. *SIAM Journal of Control and Optimization*, 34(4):1365–1391, July 1996.

D. S. Naidu. *Optimal Control Systems*. CRC Press, 2003. Chapter 5, Discrete-time optimal control systems.

R. Padhi, N. Unnikrishnan, X. Wang, and S. N. Balakrishnan. A single network adaptive critic (SNAC) architecture for optimal control synthesis for a class of nonlinear systems. *Neural Networks, Science Direct, Elsevier*, 19:1648–1660, 2006.

D. V. Prokhorov and D. C. Wunsch II. Adaptive critic designs. *IEEE Transactions on Neural Networks*, 8(5): 997–1007, September 1997.

J. Si, A. G. Barto, W. B. Powell, and D. Wunsch II, editors. *Handbook of learning and Approximate Dynamic Programming*, chapter 3. IEEE Press, 2005.

J. J. E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, New Jersey, 1991.