2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

Constrained Optimal Control for a Class of Nonlinear Systems with Uncertainties

Jie Ding, S. N. Balakrishnan, *Member, IEEE*

Abstract—**Approximate dynamic programming formulation (ADP) implemented with an Adaptive Critic (AC) based neural network (NN) structure has evolved as a powerful technique for solving the Hamilton-Jacobi-Bellman (HJB) equations. As interest in the ADP and the AC solutions are escalating, there is a dire need to consider enabling factors for their possible implementations. A typical AC structure consists of two interacting NNs which is computationally expensive. In this paper, a new architecture, called the "Cost Function Based Single Network Adaptive Critic (J-SNAC)" is presented that eliminates one of the networks in a typical AC structure. This approach is applicable to a wide class of nonlinear systems in engineering. Many real-life problems have controller limits. In this paper, a non-quadratic cost function is used that incorporates the control constraints. Necessary equations for optimal control are derived and an algorithm to solve the constrained-control problem with J-SNAC is developed. A benchmark nonlinear system is used to illustrate the working of the proposed technique. Extensions to optimal control-constrained problems in the presence of uncertainties are also considered.**

Keywords: **Approximate Dynamic Programming (ADP), Constrained Control, Optimal Control, Nonlinear Control, Cost Function Based Single Network Adaptive Critic, J-SNAC**

I. INTRODUCTION

FEEDBACK control is the preferred solution for many systems because of its beneficial properties like robustness with respect to noise and modeling uncertainties. It is well-known that a dynamic programming formulation offers the most comprehensive solution approach to nonlinear optimal control in a state feedback form (Lewis, 1992; Bryson *et al*., 1975). However, solving the associated HJB equation demands a large (rather infeasible) number of computations and storage space dedicated to this purpose. An innovative idea was proposed in (Werbos, 1992) to get around this numerical complexity by using an ADP formulation. The solution to the ADP formulation is obtained through a dual NN approach called the Adaptive Critic (AC). In one version of the AC approach, called the Heuristic Dynamic Programming (HDP), one network (called the action network) represents the mapping between the state and control variables while a second network (called the critic network) represents the mapping between the state and the cost function to be minimized. Adaptive critic formulations can be found in many papers; some researchers have used ADP formulations to solve problems with finite state spaces in applications to behavioral and computer sciences and operations research and robotics (Barto, 1991, 2004; Powell, 2004; Bertsekas, 1996). Adaptive critics can also be considered reinforcement learning designs (Barto, 1991, 2004). These formulations employ primarily cost function based adaptive critics that we consider in this paper. There are also many papers in the literature that use system science principles and neural networks to formulate the problems with applications to real-time feedback control of dynamic systems. A compendium of such applications can be found in Si *et al.* (2004). In recent years, many researchers have paid more attention on ADP in order to obtain approximate solutions of the HJB equation (Al-Tamimi. *et al*., 2008; Balakrishnan *et al*., 2008; Li and Si, 2007; Werbos, 2007). Model based synthesis of adaptive critic based controllers presented by Balakrishnan (1996), Prokhorov *et al.* (1997), Venayagamoorthy (2003) for systems driven by ordinary differential equations and Padhi *et al.* (2006) for distributed parameter systems show that the ADP based controllers stabilize the plants quite successfully. Ferrai *et al.* (2002) have implemented a global adaptive critic controller for a business jet. Yang *et al.* (2006) apply an adaptive critic based controller in an atomic force microscope based force controller to push nano particles on the substrates. Lendaris *et al.* (2000) have successfully shown in simulations that the HDP method can prevent cars from skidding when driving over unexpected patches of ice. In fact, there are many variants of the AC designs (Prokhorov *et al.*, 1997). Typically, the AC designs are formulated in a discrete framework. Hanselman *et al.* consider the use of continuous time adaptive critics in (Hanselmann *et al*., 2007).

While there has been a multitude of papers involving ACs, there is virtually no paper in the published literature that deals with the computational load (or alleviating it) associated with the AC designs. If the ADP formulation should to find their way to engineering implementations, computationally efficient AC designs that show convergence are badly needed. Balakrishnan's group (Padhi *et al*., 2004; Yadav *et al*., 2006) has proposed a Single Network Adaptive Critic (SNAC) earlier. However, that structure was based on a critic network that outputs the costates as in the DHP. The problem in dealing with costates is that they have no physical meaning in an engineering problem like the physical states of a system. Therefore, it is very difficult to get an idea of the magnitude of costates; in a multivariable

Jie Ding, Ph.D. student, Department of Mech. and Aerospace Engg., Missouri Univ. of Science and Tech., Rolla, MO 65401 USA (e-mail: jdr5c@mail.mst.edu).

S. N. Balakrishnan, Professor, Department of Mech. and Aerospace Engg., Missouri Univ. of Science and Tech., Rolla, MO 65401 USA (e-mail: bala@mst.edu).

978-1-4577-0079-8/11/$26.00 ©2011 AACC

problem, different costates can have values which could vary by several orders of magnitudes and thereby, making convergence of the critic network very difficult. In contrast, use of cost in a critic network as in this paper has much more relevance and meaning. As opposed to the costates which have the same dimension as states in a multivariable problem and therefore, demand a more diverse network structure and its training, the cost function is a ***scalar*** and therefore, the critic network dimension is minimal. The contribution of this paper is a cost based single network adaptive critic architecture. It captures the mapping between the states (at time k) and the optimal cost (from k to the end). Note that while costates have no physical meaning, the output of the cost network provides valuable information to the designer an idea of the remaining cost involved at any stage of the process as a function of the states of the system. In fact, the J-SNAC proposed in this paper is applicable to the type of control-affine nonlinear problems presented in some recent papers (Yang *et al.*, 2009; Wang *et al.*, 2009) while saving about 50% computation time as compared to the typical HDP structure used in those papers due to the elimination of one network.

Almost all real-life problems have controller limits. Bernstein (1995) developed the optimal saturating feedback control laws involving bang-bang action, which is a modification of the control laws given by Frankena and Sivan (1979), and Ryan (1982). In (Adhyaru *et al.*, 2008), an HJB equation based constrained optimal control algorithm is proposed for a bilinear system. In (Cheng *et al.*, 2006), *fixed-final time* constrained input optimal control laws using neural networks to solve Hamilton-Jacobi-Bellman equations for general affine in the input nonlinear systems are proposed. Although many methods have been proposed to deal with the constrained optimal problem, solving the associated HJB equation demands a large number of computations and storage space and this important fact should be addressed. The major contributions of this paper are that the J-SNAC technique developed in this paper 1) solves the control problem without the storage and numerical load typically associated with HJB solutions 2) presents a unifies solution for the constrained control problem through neural networks even with model uncertainties. A non-quadratic cost function (Lyshevski, 1996) is used to handle the control constraints.

Rest of the paper is organized as follows: In Section II, the ADP equations are presented and in Section III, J-SNAC technique with a non-quadratic cost function is presented. An online updated neural network is discussed in Section IV. Numerical results are presented in section V.

II. Approximate Dynamic Programming

In this section, the principles of approximate (discrete) dynamic programming, which both the AC and the J-SNAC approaches rely upon, are described. An interested reader can find more details about the derivations in (Balakrishnan *et al.*, 1996; Werbos, 1992). Note that a prime requirement to apply the AC or the J-SNAC is to formulate the problem in discrete-time. The control designer has the freedom to use any appropriate discretization scheme. For example, one can use the Euler approximation for the state equation and Trapezoidal approximation for the cost function (Gupta, 1995). In a discrete-time formulation, one wants to find an admissible control U_k, which causes the system given by

$$X_{k+1} = F_k(X_k, U_k) \qquad (1)$$

to follow an admissible trajectory from an initial point X_0 to a final desired point X_N while minimizing a desired cost function J given by

$$J = \sum_{k=0}^{N-1} \Psi_k\,(X_k, U_k) \qquad (2)$$

where $X_k \in \mathbb{R}^n$ and $U_k \in \mathbb{R}^m$ are the state and control vectors at time step k. The functions F_k and Ψ_k are assumed to be differentiable with respect to both X_k and U_k. Moreover, Ψ_k is assumed to be convex. One can notice that when $N \to \infty$, this cost function leads to a regulator (infinite time) problem. The steps in obtaining optimal control are now described.

Remark 1: It is important to note that the control U_k must both stabilize the system on a compact set, $\Omega \subset \mathbb{R}^{n\times 1}$ and make the cost functional value (2) finite so that the control is admissible (Beard 1995).

The cost function in (2) is rewritten to start from step k as

$$J_k = \sum_{\tilde{k}=k}^{N-1} \Psi_{\tilde{k}}\,(X_{\tilde{k}}, U_{\tilde{k}}) \qquad (3)$$

The cost, J_k, can be split into

$$J_k = \Psi_k + J_{k+1} \qquad (4)$$

where Ψ_k and $J_{k+1} = \sum_{\tilde{k}=k+1}^{N-1} \Psi_{\tilde{k}}$ represent the 'utility function' at time step k and the cost-to-go from time step $k+1$ to N, respectively. The $n \times 1$ costate vector at step k is

$$\lambda_k = \partial J_k / \partial X_k \qquad (5)$$

The necessary condition for optimality is given by

$$\partial J_k / \partial U_k = 0 \qquad (6)$$

Equation (6) can be further expanded as

$$\frac{\partial J_k}{\partial U_k} = \frac{\partial \Psi_k}{\partial U_k} + \frac{\partial J_{k+1}}{\partial U_k} = \frac{\partial \Psi_k}{\partial U_k} + \left(\frac{\partial X_{k+1}}{\partial U_k}\right)^T \frac{\partial J_{k+1}}{\partial X_{k+1}} \qquad (7)$$

The optimal control equation can, therefore, be written as

$$\frac{\partial \Psi_k}{\partial U_k} + \left(\frac{\partial X_{k+1}}{\partial U_k}\right)^T \frac{\partial J_{k+1}}{\partial X_{k+1}} = 0 \qquad (8)$$

The costate equation is derived in the following way

$$\lambda_k = \frac{\partial \Psi_k}{\partial X_k} + \frac{\partial J_{k+1}}{\partial X_k} = \frac{\partial \Psi_k}{\partial X_k} + \left(\frac{\partial X_{k+1}}{\partial X_k}\right)^T \frac{\partial J_{k+1}}{\partial X_{k+1}} = \frac{\partial \Psi_k}{\partial X_k} + \left(\frac{\partial X_{k+1}}{\partial X_k}\right)^T \lambda_{k+1} \qquad (9)$$

Equations (1), (8) and (9) have to be solved simultaneously, along with appropriate boundary conditions for the synthesis of optimal control. For the regulator problems, the boundary conditions usually take the form: X_0 is fixed and $\lambda_N \to 0$ as $N \to \infty$. For problems where the state equation and cost function are such that one can obtain an explicit solution for the control variable in terms of the state and the cost variables from equation (8), the J-SNAC technique is applicable.

III. J-SNAC Synthesis with Non-quadratic Cost

In this section, the cost function based single network adaptive critic (J-SNAC) technique is discussed in detail. In the J-SNAC design, the critic network captures the functional relationship between the state X_k and the cost J_k.

3.1 HJB Equation with Constraints on the Control Input

For the nonlinear system given by

$$\dot{x} = f(x) + Bu,\ x(t_0) = x_0 \quad (10)$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ are the state and control vectors, $f(\cdot) \in \mathbb{R}^n$ is the smooth mapping and $B \in \mathbb{R}^{n\times m}$. The following cost function is commonly used in the design of constrained controllers (Lyshevski, 2001):

$$J_c = \int_{t_0}^{t_f\to\infty} \left[\tfrac{1}{2}x^T Qx + \int \left(\phi^{-1}(u)\right)^T R\, du\right] dt \quad (11)$$

where $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are the diagonal weighting matrices and $\phi(\cdot)$ is the bounded, integrable, one-to-one, real-analytic globally Lipschitz continuous function $\phi \in \mathbb{R}^m$.

The Hamilton-Jacobi functional equation for (10) is

$$-\frac{\partial V}{\partial t} = \min_u \left\{\tfrac{1}{2}x^T Qx + \int \left(\phi^{-1}(u)\right)^T R\, du + \frac{\partial V^T}{\partial t}[f(x) + Bu]\right\} \quad (12)$$

where $V(\cdot)$ is the positive-definite, continuously differentiable (minimum-cost) return function, $V(x_0) = \inf_u J_c(x_0, u) > 0$. By applying the cost (11), control law can be designed as

$$u = -\phi\left(R^{-1}B^T(\partial V(x)/\partial x)\right) \quad (13)$$

In discrete-time form, the system and cost function are

$$X_{k+1} = f(X_k) + BU_k \quad (14)$$

$$J = \textstyle\sum_{k=0}^{N\to\infty} \Psi_k \quad (15)$$

where the utility function Ψ_k is given by

$$\Psi_k = \left((1/2)X_k^T Q_W X_k + M(U_k)\right)\Delta t \quad (16)$$

where $M(U_k) = \int_0^{U_k} \left(\phi^{-1}(v)\right)^T R_W dv$. By applying (8), we obtain

$$\phi^{-1}(U_k)\Delta t R_W + \left(\frac{\partial X_{k+1}}{\partial U_k}\right)^T \frac{\partial J_{k+1}}{\partial X_{k+1}} = 0 \quad (17)$$

The constrained control is obtained as

$$U_k = -\phi\left((\Delta t R_W)^{-1}B^T \frac{\partial J_{k+1}}{\partial X_{k+1}}\right) =$$
$$-\phi\left((\Delta t R_W)^{-1}B^T \left(\left(\frac{\partial X_{k+1}}{\partial X_k}\right)^T\right)^{-1} \left(\frac{\partial J_k}{\partial X_k} - \Delta t Q_W X_k\right)\right) \quad (18)$$

where it is assumed that $((\partial X_{k+1}/\partial X_k)^T)^{-1}$ exists.

The costate equation can be obtained by applying (9) as

$$\lambda_k = \Delta t Q_W X_k + [\partial X_{k+1}/\partial X_k]^T \lambda_{k+1} \quad (19)$$

3.2 Neural Network Training

In the J-SNAC, the steps for the training the critic network, are as follows (Fig. 1):

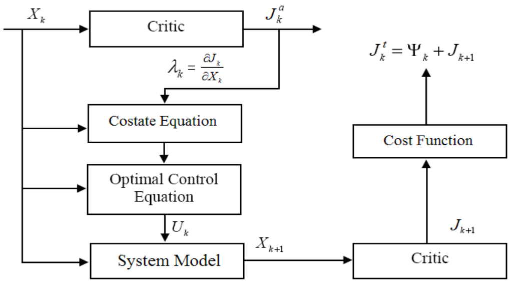

Fig. 1. J-SNAC Network Training Scheme

a. Input X_k to the critic network to obtain $J_k = J_k^a$.

b. Calculate $\lambda_k = \partial J_k/\partial X_k$, and λ_{k+1} by equation (19)

c. Calculate U_k, form the optimal control equation (18).

d. Use X_k and U_k to get X_{k+1} from equation (1).

e. Input X_{k+1} to the critic network to get J_{k+1}.

f. Use X_k, U_k and J_{k+1}, to calculate J_k^t with equation (4).

g. Train the critic network by solving equation (A.7) for network weights (See Appendix).

h. Check the convergence of the critic network (by defining the relative error $e_c \equiv (\|J_k^t - J_k^a\|/\|J_k^t\|)$, the training process is stopped when $\|e_c\| < tol_c$, otherwise, repeat steps *a*-*g*).

A numerical method for J-SNAC training is presented in the Appendix.

IV. Dynamic Re-optimization of J-SNAC

In this section, we consider the plant dynamics with parametric uncertainties or unmodeled nonlinearities. We discuss the dynamic re-optimization of the J-SNAC controller in response to the model changed due to the uncertainties. This is achieved with a virtual plant model that is similar to the actual plant but has a term to capture the uncertainties with an online neural network. Consider a general nonlinear system given as

$$X_{k+1} = f(X_k) + BU_k \quad (20)$$

where $X_k \in \mathbb{R}^n$ is the state vector and $U_k \in \mathbb{R}^m$ is the control vector. Let the actual plant have the structure

$$X_{k+1} = f(X_k) + BU_k + d(X_k) \quad (21)$$

where the controller U_k will have to be re-optimized to optimize the plant performance with the unmodeled dynamics $d(X_k)$ present. Since the term $d(X_k)$ in the plant equation is unknown, the first step in controller re-optimization is to approximate the uncertainty in the plant equation. For this purpose a virtual plant is defined. Let X_a represent the state vector of the virtual plant.

The dynamics of this virtual plant is governed by

$$X_{a_{k+1}} = f(X_k) + BU_k + \hat{d}(X_k) + K\left(X_k - X_{a_k}\right), X_a(0) = X(0) \quad (22)$$

where $K > 0$ is a design parameter. We assume that we have all the actual plant states, X, available for measurement at every step. The term $\hat{d}(X_k)$ is the neural network approximation of the actual plant. Subtracting equation (22) from (21), by defining $E_{a_k} \equiv X_{a_k} - X_k$, we obtain $X_{a_{k+1}} - X_{k+1} = \hat{d}(X_k) - d(X_k) + K\left(X_k - X_{a_k}\right)$ or $E_{a_{k+1}} = \hat{d}(X_k) - d(X_k) - KE_{a_k}$. It can be seen that as $\hat{d}(X_k) - d(X_k)$ approaches zero, E_a is exponentially stable, *i.e.* $E_a \to 0$ as $t \to \infty$. The J-SNAC dynamic re-optimization scheme is shown in Fig. 2.

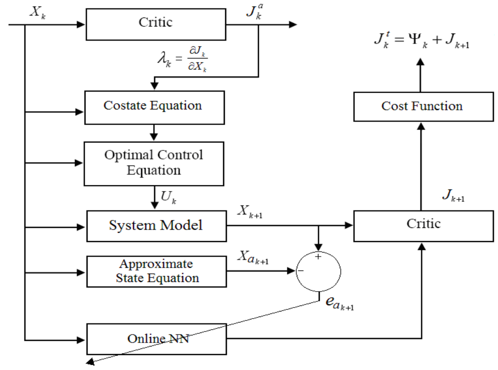

Fig. 2. J-SNAC Dynamic Re-optimization Scheme

Defining $X = [x_1, x_2, \cdots, x_n]^T$, $E_a = [e_{a1}, e_{a2}, \cdots, e_{an}]^T$, and $d(X) \equiv [d_1(X), d_2(X), \cdots, d_n(X)]^T$, where $d_i(X)$ denotes the unmodeled dynamics in the differential equation for the i^th state of the system. The approach in this study is to have 'n' NNs, one for each component of the unmodeled dynamics, that accommadtes simpler development and analysis. (See Fig. 3). The state equation for each channel is given by

$$x_{i_{k+1}} = f_i(X_k) + b_i U_k + d_i(X_k) \quad (23)$$

$$x_{ai_{k+1}} = f_i(X_k) + b_i U_k + \hat{d}_i(X_k) + e_{ai_k} \tag{24}$$

where $e_{ai_k} \equiv x_{ai_k} - x_{i_k}$. Subtracting (24) from (23), we obtain

$$x_{ai_{k+1}} - x_{i_{k+1}} = \hat{d}_i(X_k) - d_i(X_k) - Ke_{ai_k} \tag{25}$$

How do we approximate the uncertainty $d_i(X_k)$ with a neural network? Different architectures for neural networks exist in the literature (Haykin, 1999). In this chapter, a linear-in-the-parameter network is chosen. The reasons are two fold: keep the architecture simple and mathematically tractable. Let us assume that there exists an NN with an optimum set of weights that approximates $d_i(X)$ within a certain accuracy of ε_i. Thus we have

$$d_i(X_k) = W_{i_k}^T \varphi_i(X_k) + \varepsilon_i. \tag{26}$$

In equation (26), $\varphi_i(\cdot)$ represents basis functions used in the neural network approximations. Choosing proper basis functions for a given problem is still an art and not a science. In applications, one examines the system model and selects the number and form of the basis functions from the states of the system. Note that $\hat{d}_i(X_k) = \widehat{W}_{i_k}^T \varphi_i(X_k)$, where $\widehat{W}_{i_k}^T \varphi_i(X_k)$ is the output of the uncertainty approximation NN. $\widehat{W}_{i_k}$ represents the uncertainty approximation NN weights. Substituting (26) into (25), we obtain

$$x_{ai_{k+1}} - x_{i_{k+1}} = \widehat{W}_{i_k}^T \varphi_i(X_k) + \varepsilon_i - W_{i_k}^T \varphi_i(X_k) - e_{ai_k} \tag{27}$$

$$e_{ai_{k+1}} = x_{ai_{k+1}} - x_{i_{k+1}} = \widetilde{W}_{i_k}^T \varphi_i(X_k) + \varepsilon_i - e_{ai_k} \tag{28}$$

where $\widetilde{W} = W_i - \widehat{W}_i$, is the difference between the optimal weights of the NN that represents $d_i(X_k)$ and the actual network weights. More details on the update rule can be found in (Padhi 2007; Unnikrishnan *et al.* 2006).

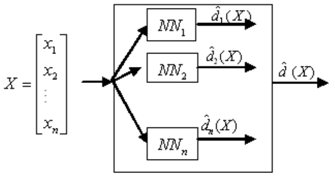

Fig. 3. Uncertainty Neural Network Structure

V. NUMERICAL RESULTS

5.1 Example: Spacecraft Control

5.1.1. Problem Description and Optimality Conditions

The dynamic rotational motion equation (Slotine and Li, 1991) is given by

$$I\dot{\omega} = p \times \omega + \tau \tag{29}$$

where I is the matrix of moment of inertias, ω is the angular velocity, p is the total spacecraft angular momentum expressed in spacecraft coordinates and τ is the torque applied to the spacecraft by the reaction wheels motors.

Choosing the states $x = [\phi, \theta, \psi]^T$, the kinematic equations describing the attitude of a spacecraft may be written as

$$\dot{x} = J(x)\omega \tag{30}$$

where

$$J(x) = \begin{bmatrix} 1 & \sin(\phi)\tan(\theta) & \cos(\phi)\tan(\theta) \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi)\sec(\theta) & \cos(\phi)\sec(\theta) \end{bmatrix} \tag{31}$$

The total spacecraft angular momentum p is written as

$$p = R(x)p^I \tag{32}$$

where $p^I = [1, -1, 0]^T$ is the (constant) inertial angular momentum and $R(x)$ can be found in (Slotine and Li, 1991).

Choosing $X = [\phi, \theta, \psi, \omega_x, \omega_y, \omega_z]^T$ as the states, $U = [\tau_1, \tau_2, \tau_3]^T$ as the control, and assuming an uncertainty $d = [d_1, d_2, 0]^T$ to be present in the system, the dynamics of the system are characterized by

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = \begin{bmatrix} 0_{3\times3} & J(x) \\ 0_{3\times3} & I^{-1}p\,\times \end{bmatrix} \begin{bmatrix} \phi \\ \theta \\ \psi \\ \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} + \begin{bmatrix} 0_{3\times3} \\ I^{-1} \end{bmatrix} \begin{bmatrix} \tau_1 \\ \tau_2 \\ \tau_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ d_1 \\ d_2 \\ 0 \end{bmatrix} \tag{33}$$

where $d_1 = 0.1\omega_x^3, d_2 = 0.5\sin(\omega_x)\cos^2(\omega_y)$ are uncertainties. The control objective is to drive all the states to zero as $t \to \infty$. A non-quadratic cost function, J_c, is selected as

$$J_c = \int_0^\infty \left[\frac{1}{2}X^T Q_W X + \int_0^U \left(\phi^{-1}(U_{max}^{-1}U)\right)^T R_W\, dU\right] dt \tag{34}$$

where $Q_W \geq 0$ and $R_W > 0$ are weighting matrices for state and control respectively. U_{max} is the control constraint.

The state equation is discretized as

$$X_{k+1} = X_k + \Delta t[f(X_k) + BU_k] \tag{35}$$

The non-quadratic cost function (34) is discretized as

$$J_d = \sum_{k=0}^{N\to\infty} \left(\frac{1}{2}{X_k}^T Q_W X_k + \int_0^{U_k} \left(\phi^{-1}(U_{max}^{-1}U_k)\right)^T R_W\, dU_k\right) \Delta t \tag{36}$$

Optimality condition leads to the control equation

$$\begin{aligned} U_k &= U_{max}\phi\left(-(\Delta t R_W)^{-1} B^T (\partial J_{k+1}/\partial X_{k+1})\right) \\ &= U_{max}\phi\left(-(\Delta t R_W)^{-1} B^T \left(\left(\frac{\partial X_{k+1}}{\partial X_k}\right)^T\right)^{-1} \left(\frac{\partial J_k}{\partial X_k} - \Delta t Q_W X_k\right)\right) \end{aligned} \tag{37}$$

The costate equation can be obtained as

$$\lambda_k = \Delta t Q_W X_k + [\partial F_k / \partial X_k]^T \lambda_{k+1} \tag{38}$$

where F_k represents the expression on the right hand side of equation (35). For this problem, $\Delta t = 0.005, Q_W$ is selected as $diag(20, 20,20,0,0,0)$ and R_W is selected as $diag(10^{-3}, 10^{-3}, 10^{-3})$. The Lipschitz continuous function ϕ is selected as $\phi(\cdot) = \tanh(\cdot)$. In the J-SNAC synthesis, the cost J_k is a function of the network weights $\widehat{W}^k$ and states X_k, which is given by equation (A.2) as $J_k = {\widehat{W}^k}^T \Phi(X_k)$. In this problem, the network weights were initialized to zero. The basis function is selected as $\begin{bmatrix} X_1, \dots, X_6, X_1^2, \dots, X_6^2, X_1X_2, X_1X_3, X_1X_4, X_1X_5, X_1X_6, X_2X_3, \\ X_2X_4, X_2X_5, X_2X_6, X_3X_4, X_3X_5, X_3X_6, X_4X_5, X_4X_6, X_5X_6 \end{bmatrix}$ and m, the number of sets in (A.7) for the network synthesis is selected as 100.

5.1.2. Uncertainty Estimation

In this study, the uncertainty NN structure is $\hat{d}_i = W_i^T \varphi$, $i = 1, 2$. The observer gain K is selected as 10. W_1 and W_2 are both initialized as 27×1 random vectors. The basis functions is selected as $\varphi = kron(kron(c_1, c_2), c_3)$, where $c_1 = [1\ \sin(\omega_x)\ \cos(\omega_x)]^T, c_2 = \left[1\ \sin(\omega_y)\ \cos(\omega_y)\right]^T, c_3 = [1\ \sin(\omega_z)\ \cos(\omega_z)]^T$, and $kron$ denotes Kronecker product.

The discretized equations can be written as

$$X_{k+1} = X_k + \Delta t[f(X_k) + BU_k + d] \tag{39}$$

Expression for optimal control is the same as equation (37). The costate equation though changes to

$$\lambda_k = \Delta t Q_W X_k + [\partial F_k' / \partial X_k]^T \lambda_{k+1} \tag{40}$$

where F_k' represents the expression on the right hand side of equation (39), which also accounts for the uncertainty. During each iteration of the simulation, the critic network is updated. Since d_k is unknown, $\hat{d}_k$, the output of the online neural network is used. The online training is carried out at every instant k by using E_k as the input to the cost network and obtaining the new target cost J_k^t as the output.

]

5.1.3. Analysis of Results

Actual and reference state histories are shown in Fig. 4 and Fig. 5. It can be seen that all the states go to zero within 5 seconds. Fig. 6 shows the history of the constrained control. The control limit is selected as follows: firstly the program is run without control constraint to find the maximum control value U_m, which is $U_m = [\tau_{1m}, \tau_{2m}, \tau_{3m}]^T = [71,71,69]^T\ (N \cdot m)$ and then to demonstrate the working of the proposed constrained optimal control technique, the control constraint U_{max} is selected as 50% of U_m. It can be seen that at the very beginning the torque stays at $U_{max} = 0.5U_m = [35.5,35.5,34.5]^T\ (N \cdot m)$ and is within the limit after that.

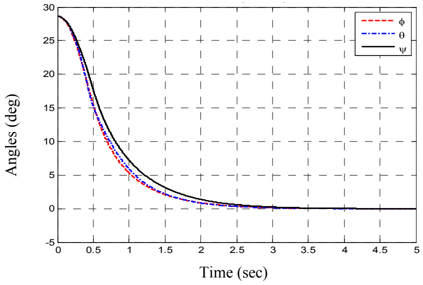

Fig. 4. Histories of Angles

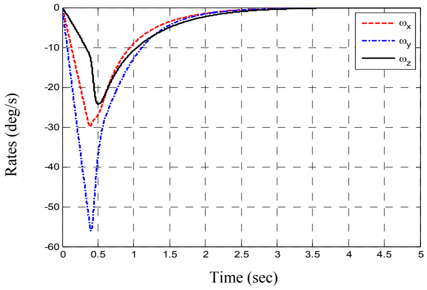

Fig. 5. Histories of Rates

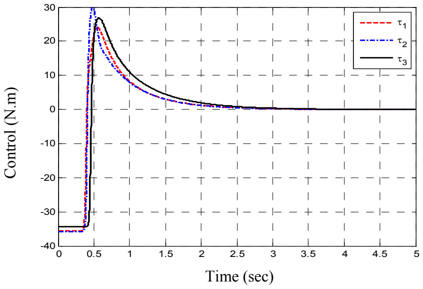

Fig. 6. Constrained Control History

True and estimated uncertainty histories are shown in Fig. 7. It can be seen that the estimated uncertainties nicely and quickly track the true uncertainties. Weights histories of the uncertainty NNs are shown in Fig. 8 and Fig. 9.

VI. Conclusions

In this paper, an online J-SNAC technique has been presented to solve nonlinear constrained control problems with model uncertainties. A non-quadratic cost function is used that incorporates the control constraints. Necessary equations for optimal control are derived and an algorithm to solve the constrained-control problem with J-SNAC is developed. Extensions to optimal control-constrained problems in the presence of uncertainties are also considered. A spacecraft control problem is used to illustrate the working of the proposed technique.

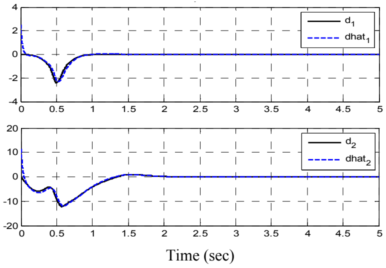

Fig. 7. True and Estimated Uncertainties Histories

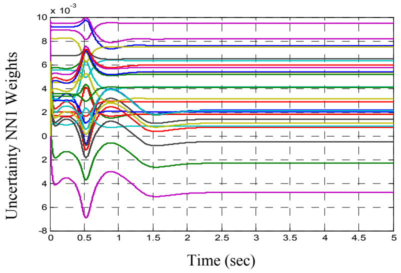

Fig. 8. Weights Histories

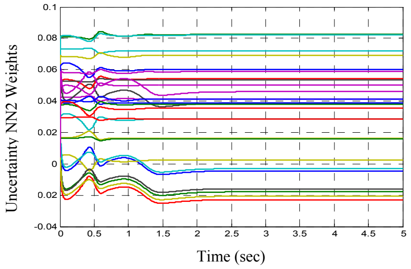

Fig. 9. Weights Histories

Appendix

This appendix is derived from the convergence proof by Al-Tamimi *et al.* (2008). The difference is that the action network is eliminated in this study. Consider a discrete nonlinear control-affine system

$$X_{k+1} = f(X_k) + BU_k \qquad (A.1)$$

where X_k is an $n \times 1$ vector, U_k is an $m \times 1$ vector, $f(\cdot)$ can be a nonlinear function of the states, B is a constant $n \times m$ matrix.

J_k is a function of the network weights $\hat{W}^k$and states X_k

$$J_k = \hat{W}^{k^T}\Phi(X_k) \qquad (A.2)$$

$$\lambda_k = \partial\left(\hat{W}^{k^T}\Phi(X_k)\right)/\partial X_k \qquad (A.3)$$

where it is assumed that X_k stay within a compact domain, $X_k \subset \Omega_c$ so that $\partial\phi(X_k)/\partial X_k$ is bounded.

In i^{th} iteration, if a non-quadratic cost function in (11) is used, the control is computed as

$$U_k^i = \phi\left(-R^{-1}B^T\left(\left(\frac{\partial f(X_k)}{\partial X_k}\right)^T\right)^{-1}\left(\frac{\partial \hat{W}^{i^T}\Phi(X_k)}{\partial X_k} - QX_k\right)\right) \qquad (A.4)$$

Cost relationship at stages k and $k+1$ is given by

$$J_k^i = \Psi_k^i + J_{k+1}^i \qquad (A.5)$$

Substituting equation (A.2) into equation (A.5), we obtain

$$\hat{W}^{i+1^T}\Phi(X_k) = \Psi_k^i + \hat{W}^{i^T}\Phi(X_{k+1}) \qquad (A.6)$$

Equation (A.6) is linear in $\hat{W}^{i+1}$ with m unknowns, where m is number of elements of $\Phi(X_k)$. Taking the transpose of equation (A.6), and selecting m sets of states X_k called $X_k^{(1)}$ to $X_k^{(m)}$, it ends up with m equations with m unknowns:

$$\begin{cases} \Phi\left(X_k^{(1)}\right)^T \widehat{W}^{i+1} = \Psi_k^{i,1} + \Phi(X_{k+1}^{(1)})^T \widehat{W}^i \\ \vdots \\ \Phi\left(X_k^{(m)}\right)^T \widehat{W}^{i+1} = \Psi_k^{i,m} + \Phi(X_{k+1}^{(m)})^T \widehat{W}^i \end{cases} \qquad (A.7)$$

where for $j = 1, 2, \ldots, m$.

$$U_k^{i,(j)} \equiv U^i\left(X_k^{(j)}\right) = \phi\left(-R^{-1}B^T A^{-1}\left(\frac{\partial \widehat{W}^{i^T} \Phi\left(X_k^{(j)}\right)}{\partial X_k^{(j)}} - QX_k^{(j)}\right)\right) \qquad (A.8)$$

$$X_{k+1}^{(j)} = f\left(X_k^{(j)}\right) + BU_k^{i,(j)} \qquad (A.9)$$

Equations (A.7) can be rewritten as

$$\mathbf{\Phi}(X_k)\widehat{W}^{i+1} = RHS\left(X_k, \widehat{W}^i\right) \qquad (A.10)$$

where the *RHS* is the $m \times 1$ vector composed of all the *RHS* of equation (A.7) and the $m \times m$ matrix $\mathbf{\Phi}(X_k)$ is given by

$$\mathbf{\Phi}(X_k) = \begin{bmatrix} \Phi\left(X_k^{(1)}\right)^T \\ \vdots \\ \Phi\left(X_k^{(m)}\right)^T \end{bmatrix}, X_k = \begin{bmatrix} X_k^{(1)^T} \\ \vdots \\ X_k^{(m)^T} \end{bmatrix} \qquad (A.11)$$

By using (A.11) in (A.10), a recursive relationship for the network weights is given as

$$\widehat{W}^{i+1} = \mathbf{\Phi}(X_k)^{-1} RHS\left(X_k, \widehat{W}^i\right) \qquad (A.12)$$

For the inverse $\mathbf{\Phi}(X_k)^{-1}$ to exist, $X_k^{(j)}$'s should not be identical and the elements of vector $\phi(X_k)$ should be linearly independent. One can select more than 'm' minimum required sets of states, and formulate a recursive relationship for the over-defined system of equations. In this case, the unique solution of the least squares minimization problem is:

$$\widehat{W}^{i+1} = \left(\mathbf{\Phi}(X_k)^T \mathbf{\Phi}(X_k)\right)^{\mathbf{-1}} \mathbf{\Phi}(X_k)^T RHS\left(X_k, \widehat{W}^i\right) \qquad (A.13)$$

Acknowledgment

Support for this study from NASA under Grant No: ARMD NRA NNH07ZEA001N-IRAC1 and the National Science Foundation are gratefully acknowledged. The views of the authors do not necessarily represent the views of the NSF and NASA.

References

[1] D. M. Adhyaru and I. N. Kar, "Constrained Optimal Control of Bilinear Systems Using Neural Network Based HJB Solution," *2008 International Joint Conf. on Neural Networks*, pp.4137-4142, 2008.

[2] A. Al-Tamimi, F. L. Lewis, and M. Abu-Khalaf, "Discrete-time Nonlinear HJB Solution Using Approximate Dynamic Programming: Convergence proof," *IEEE Trans. Syst.,Man., Cybern. B*, vol. 38, no. 4, pp. 943-949, 2008.

[3] S. N. Balakrishnan, J. Ding, and F. L. Lewis. "Issues on Stability of ADP Feedback Controllers for Dynamical Systems," *IEEE Trans. Syst., Man., Cybern. B*, vol. 38, no. 4, pp. 913–917, 2008.

[4] S. N. Balakrishnan, and V. Biega, "Adaptive-Critic Based Neural Networks for Aircraft Optimal Control," *Journal of Guidance, Control and Dynamics*, vol. 19, pp. 893-898, 1996.

[5] A. G. Barto, "Connectionist Learning for Control," in *Neural Networks for Control.* Cambridge, MA: MIT Press, 1991.

[6] A. G. Barto and T. Dieterich, "Reinforcement Learning and its Relation to Supervised Learning," in *Handbook of Learning and Approximate Dynamic Programming*. New York: Wiley-IEEE Press, Aug. 2004.

[7] R. W. Beard, "Improving the Closed-loop Performance of Nonlinear Systems," Ph.D. Thesis, Rensselaer Polytechnic Institute, 1995.

[8] D. S. Bernstein, "Optimal Nonlinear, but Continuous, Feedback Control of Systems with Saturating Actuators," *International Journal of Control*, Vol.62, No.5, pp.1209-1216, 1995.

[9] D. P. Bertsekas and J. N. Tsitsiklis, Neuro-Dynamic Programming, Belmont, MA: Athena Scientific, 1996.

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

[11] T. Cheng and F. L. Lewis, "Fixed-Final Time Constrained Optimal Control of Nonlinear Systems Using Neural Network HJB Approach," *Proc. of the 45th IEEE Conf. on Decision & Control*, San Diego, CA, pp.3016-3021, 2006.

[12] S. Ferrari and R. Stengel, "An Adaptive Critic Global Controller," *Proc. Amer. Control Conf.*, pp.2665-2670, 2002.

[13] J. F. Frankena and R. Sivan, "A Non-linear Optimal Control Law for Linear Systems," *Int. J. Control*, Vol.30, pp.159-178, 1979.

[14] S. K. Gupta, Numerical Methods for Engineers, *Wiley Eastern Ltd. and New Age International Ltd,* 1995.

[15] T. Hanselmann, L. Noakes, and A. Zaknich, "Continuous Time adaptive Critics," *IEEE Transactions on Neural Networks,* Vol. 18, Issue 3, pp. 631 – 647, 2007.

[16] S. Haykin, Neural Networks: A Comprehensive Foundation, *Prentice Hall*, 1999.

[17] G. Lendaris, L. Schultz, T. Shannon, "Adaptive Critic Design for Intelligent Steering and Speed Control of a 2-axle Vehicle," *Proc. International Joint Conf. on Neural Networks*, Como, Italy, 2000.

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

[19] B. Li and J. Si, "Robust Dynamic Programming for Discounted Infinite- horizon Markov Decision Processes with Uncertain Stationary Transition Matrices," *Proc. IEEE Int. Symp. Appr.Dynamic Prog. and Reinforcement Learning*, Honolulu, HI, pp. 96-102, 2007.

[20] S. E. Lyshevski, "Constrained Optimization and Control of Nonlinear Systems: New Results in Optimal Control," *Proceedings of the 35th Conference on Decision and Control*, Kobe, Japan, 1996.

[21] S. E. Lyshevski, Control Systems Theory with Engineering Applications, Birkhäuser, 2001.

[22] R. Padhi, and S. N. Balakrishnan, "Optimal Beaver Population Management Using Reduced Order Distributed Parameter Model and Single Network Adaptive Critics," *Proc. Amer. Ctrl. Conf.*, Boston, MA, pp.1598-1603, 2004.

[23] R. Padhi, N. Unnikrishnan, S. N. Balakrishnan, "Model-following Neuro-adaptive Control Design for Non-square, Non-affine Nonlinear Systems," *IET Theory Appl.*, Vol.1 (6), pp.1650-1661, 2007.

[24] W. Powell and B. Van Roy, "ADP for High-dimensional Resource Allocation Problems," in *Handbook of Learning and Approximate Dynamic Programming*. New York: Wiley-IEEE Press, Aug. 2004.

[25] D. Prokhorov and D. Wunsch II, "Adaptive Critic Designs," *IEEE Transactions on Neural Networks*,vol. 8, pp.997-1007, 1997.

[26] E. P. Ryan, "Optimal Feedback Control of Saturating Systems," *Int. J. Control*, Vol. 35, pp.521-534, 1982.

[27] J. Si, P. A. G. Barto, and W. B. D. Wunsch, Eds., Handbook of Learning and Approximate Dynamic Programming, *IEEE Press Series on Computational Intelligence*, New York: Wiley-IEEE Press, 2004.

[28] J.-J. E. Slotine and W. Li, Applied Nonlinear Control, *Prentice-Hall*, cha. 9, 1991.

[29] N. Unnikrishnan, S. N. Balakrishnan, R. Padhi, "Dynamic Re-optimization of a Spacecraft Attitude Controller in the Presence of Uncertainties," *Proc. the 2006 IEEE International Symposium on Intelligent Control*, Munich, Germany, pp.452-457, 2006.

[30] G. Venayagamoorthy, R. Harley, D. Wunsch, "Dual Heuristic Programming Excitation Neurocontrol for Generators in a Multimachine Power System," *IEEE Trans. Ind. Appl.*, vol.39, pp. 382-384, 2003.

[31] F. Wang, H. Zhang, D. Liu, "Adaptive Dynamic Prog. An Intro.," *IEEE Computational Intelligence Magazine*, pp.39-47, 2009.

[32] P. J. Werbos, "Approximate Dynamic Programming for Real-time Control and Neural Modeling," *Handbook of Intell. Ctrl., Multiscience Press,* 1992.

[33] P. J. Werbos, "Using ADP to Understand and Replicate Brain Intelligence: the Next Level Design," *Proc. IEEE Symp. Appr. Dynamic Programming and Reinforcement Learning*, Honolulu, HI, pp. 209-216, 2007.

[34] V. Yadav, R. Padhi, S. N. Balakrishnan, "Robust/Optimal Temperature Profile Control Using Neural Networks," *Proc. IEEE International Conf. on Ctrl. Applications*, Munich, Germany, pp.3169-3174, 2006.

[35] Q. Yang and S. Jagannathan, "Adaptive Critic Neural Network Force Controller for Atomic Force Microscope-based Nanomanipulation," *Proc. IEEE Int. Symp. Intell. Ctrl.*, pp. 464-469, 2006.

[36] L. Yang, J. Si, K. S. Tsakalis, A. A. Rodriguez, "Direct Heuristic Dynamic Programming for Nonlinear Tracking Control with Filtered Tracking Error," *IEEE Trans. on Systems, Man, and Cybernetics-Part B: Cybernetics*, Vol.39, No.6, pp.1617-1622, 2009.