ROBUST ADAPTIVE CRITIC BASED NEUROCONTROLLERS
FOR SYSTEMS WITH INPUT UNCERTAINTIES

Zhongwu Huang* and S.N.Balakrishnan**

zhuang@umr.edu bala@umr.edu

Department of Mechanical and Aerospace Engineering and Engineering Mechanics
University of Missouri-Rolla
Rolla, MO. 65401, USA

ABSTRACT

A two-neural network approach to solving nonlinear optimal control problems is described in this study. This approach called the adaptive critic method consists of one neural network called the supervisor or the critic and a second network called an action network or a controller. The inputs to both these networks are the current states of the system to be controlled. Targets for each network updates are obtained with outputs of the other network, state propagation equations and the conditions for optimal control. When their outputs are mutually consistent, the controller network output is optimal. The optimality is however limited by the underlying system model. Hence, we develop a Lyapunov based theory for robust stability of these controllers when there is input uncertainty. This results in an expression for extra control. This extra control added with the base control effort keeps the system stable under input uncertainties. We illustrate this approach through a longitudinal autopilot design of a nonlinear missile problem.

1. INTRODUCTION

Outside of dynamic programming [1], currently there is no unified mathematical formalism under which a controller can be designed for nonlinear systems. Techniques like feedback

* Ph.D student
** Professor
linearization have been used for a few nonlinear problems under limited conditions, such as equal number of inputs and outputs. This methodology is based on the idea that if a state or control transformation can be found, the original nonlinear system can be transformed to a linear system, we can design the controller for the resulting linear system with some standard method; we can then transform the control and state back to original coordinates via an inverse transformation; the net result will be a nonlinear controller. The conditions for the validity of such transformations have been studied in detail [2,3,4]. A particular case of feedback linearizing control, called dynamic inversion, has been studied at great detail for application to supermaneuverable aircraft [5,6,7]. However, dynamic inversion is sensitive to modeling errors.

In addition to feedback linearization, there are also other methods for nonlinear control design. Chen and Narendra [8] have proposed a design method by the Nonlinear Auto-Regressive Moving-Average (NARMA). Sontag[9,10] developed an approach called input-to-state stabilization. There are also methods based on \( H^\infty \) control[11,12,13] and formulations such as backstepping based on Lyapunov theorems[14].

A unique universal method to deal with control of any system, linear or nonlinear is the optimal control theory. Optimization has been a field of interest to mathematicians, scientists and engineers for a long time. Problems of optimization of functions or functionals and optimal control of linear or nonlinear dynamical systems can be solved through direct or indirect methods [15]. In direct methods where, in general, the cost function is evaluated or indirect methods where, in general, values of the derivatives are used to check optimum, separate solutions are obtained for each set of parameters or initial conditions. The difficulty with optimization of nonlinear systems is having to deal with Lagrange’s multipliers (called costates
in some studies). Unlike the physical variables associated with any given problem, it is hard to have an intuitive idea about the magnitudes of the costates; furthermore, they are known (if at all) at the final time only. As a result, many studies have focused on approximations to the underlying optimization problems or avoid having to deal with them through using Ricatti variables as done in typical feedback control problems with a quadratic cost function. In this study, we propose a formulation which alleviates the difficulties with the Lagrange’s multipliers by a backward sweep approach. Such a formulation is possible using neural networks.

Neural adaptive control, where neural nets are used instead of linear mapping in standard adaptive control, is very popular. One of the earliest papers was published by Narendra and Parthasarathy[16]. An adaptive neurocontroller with guaranteed stability was developed by Polycarpou and Ioannou [17] under certain assumptions. Further research in neural adaptive control has been presented by Sanner and Slotine[18], Lewis, et al., [19]. Calise, et al. [20,21,22,23] use a single layer neural network for adaptive control of feedback linearized aircraft model. Kim [21] developed an architecture in which neural networks approximately cancel residual nonlinearities remaining after feedback linearization. McFarland [24] has extended this technique to a tracking problem with input uncertainty. He has shown how an on-line updated neural network can stabilize an autopilot where input uncertainty exists. McFarland’s paper forms the basis for the numerical example used in this paper.

The adaptive critic design is more complex than the neurocontrol methods used in many studies. Adaptive critic design has its origin in reinforcement learning, see, for example Barto, Sutton and Williams [25]. In this paper, however, it is used as a computational tool to solve optimal control problems based on approximate dynamic programming. This formulation 1)
solves a nonlinear control problem directly without any approximation to the system model (in the absence of a good model this approach can synthesize a nonlinear model of the states), 2) yield a control law in a feedback form as a function of the current states, and 3) maintain the same structure regardless whether the problem is linear or nonlinear although the network structure may be different.

The reason for choosing this structure for formulating the optimal control problems is that this approach needs NO external training as in other forms of neurocontrollers, this is not an open loop optimal controller but a feedback controller. Balakrishnan and Biega [26] have shown the usefulness of this architecture for infinite/finite-time linear problems.

The method discussed in this study determines an optimal control law for a system by successively adapting two networks, an action network and a critic network. This method determines the control law for an entire range of initial conditions [26]. It simultaneously determines and adapts the neural networks to the optimal control policy for both linear and nonlinear systems. Further contribution of this paper is the development of theory to obtain robustness results with regard to the adaptive critic based neurocontrollers there has been (to our knowledge) no reported results in this important area. In particular, we consider uncertain dynamics associated with inputs. There has been some work in this area. Krstic [27] et al. has developed a technique for control of feedback linearizable systems with input unmodeled dynamics; McFarland [28] has used the method of dynamic nonlinear damping for a neural network based controller. However, his results are based on an inversion-based neurocontroller. In this paper, the results section clearly shows that the adaptive critic based neurocontrollers provide excellent performance under input uncertainty.
We present the general formulation of adaptive critic in Section 2; theory related to the robustness of these controllers is presented in Section 3. Numerical results are presented in Section 4 and the conclusions are drawn in Section 5.

2. PROBLEM FORMULATION AND SOLUTION DEVELOPMENT

2.1 DYNAMIC PROGRAMMING

Dynamic programming method provides a computational technique to apply the principle of optimality to a sequence of decisions which define an optimal control policy. A general mathematical description of the optimality conditions is the Bellman’s equation given by

$$J(x(k)) = \min_{u(k)} [J(x(k + 1)) + U(x(k),u(k))]$$  \hspace{1cm} (1)

In Eq. (1), state at time step $k$ is given by $x(k)$ and the control by $u(k)$. $J(x(k))$ represents the minimum cost associated with going from step $k$ to final step. $U(x(k),u(k))$ is the utility function denoting the cost from $k$ to $k+1$ using control $u(k)$ and $J(x(k + 1))$ is the minimum cost associated with going from step $k+1$ to final step.

Define co-state $\lambda(x(k))$ as

$$\lambda(x(k)) = \frac{\partial J(x(k))}{\partial x(k)}$$ \hspace{1cm} (2)

Eq. (2) can also be written in a backwards form as

$$\lambda(x(k)) = \frac{\partial U(x(k),u(k))}{\partial x(k)} + \frac{\partial u(k)}{\partial x(k)} \frac{\partial U(x(k),u(k))}{\partial u(k)} + \left[ \frac{\partial x(k + 1)}{\partial x(k)} + \frac{\partial u(k)}{\partial x(k)} \frac{\partial x(k + 1)}{\partial u(k)} \right] \lambda(x(k + 1))$$ \hspace{1cm} (3)

From (1), the Bellman’s optimality equations are given by

$$\frac{\partial J(x(k))}{\partial u(k)} = \frac{\partial x(k + 1)}{\partial u(k)} \lambda(x(k + 1)) + \frac{\partial U(x(k),u(k))}{\partial u(k)} = 0$$ \hspace{1cm} (4)
Eq. (3) and (4) should be solved in conjunction with the underlying dynamic model for the optimal policy.

For our problem, the utility function $U(x(k), u(k))$ is the popularly used quadratic function,

$$U(x(k), u(k)) = \frac{1}{2} [x^T(k)Qx(k) + u^T(k)Ru(k)]$$  \hspace{1cm} (5)

The underlying system model is given by

$$x(k+1) = f(x(k), u(k))$$  \hspace{1cm} (6)

where $f(.)$ can be either linear or nonlinear.

For the convenience, $U(x(k), u(k))$ is denoted as $U(k)$, $\lambda(x(k))$ is denoted as $\lambda(k)$ and $J(x(k))$ is denoted as $J(k)$ in the following sections.

### 2.2 ADAPTIVE CRITIC

Adaptive critic methodology has been proposed as a new optimization technique combining together concepts of reinforcement learning and dynamic programming. The goal is to find the control which minimizes the cost in Eq. (1) by solving Eqs. (3) and (4) with the use of Eq. (6) and the known initial states. In order to accomplish this task, we use two networks as in figure 1 and 2. One network called ‘action’ models the control, $u(k)$, for which the inputs are the current states, $x(k)$. The other called ‘critic’ models the co-state $\lambda(k)$, for which the inputs are also $x(k)$. In order to train the critic network as in figure 1, first, $x(k)$, is randomized and input to the action network to output $u(k)$. The system model in Eq. (6) is then used to find $x(k+1)$.

The derivative $\frac{\partial x(k + 1)}{\partial u(k)}$, $\frac{\partial x(k + 1)}{\partial x(k)}$, $\frac{\partial U(k)}{\partial u(k)}$ and $\frac{\partial U(k)}{\partial x(k)}$ can be calculated since $x(k)$ and $u(k)$ are known. Now, a randomized critic network is considered and $\lambda(k)$ and $\lambda(k + 1)$ are
calculated corresponding to $x(k)$ and $x(k+1)$. With $\lambda(k+1)$, the target $\hat{\lambda}(k)$, denoted by $\hat{\lambda}^*(k)$, can be calculated by using Eq. (3). The difference between $\hat{\lambda}^*(k)$ and $\hat{\lambda}(k)$ is used to correct the critic network. After the critic network has converged, we use this critic or supervisory network to correct the action network as in figure 2. This is done by finding $u(k)$ for random $x(k)$ and correcting them through the use of the model equation in Eq. (6) to find $x(k+1)$, and by using $x(k+1)$ to find $\hat{\lambda}(k+1)$ from the critic network corresponding to $x(k+1)$. By using $\hat{\lambda}(k+1)$ in Eq. (4), we can solve for the target $u^*(k)$ and use it to correct the action network. This two-step procedure continues till a predetermined level of convergence is reached. When these two network outputs are mutually consistent, the action network outputs optimal control. Convergence of this iterative process has been studied by Liu [29].

3. ROBUST DESIGN

3.1 PROBLEM DESCRIPTION

Consider a nominal nonlinear system (with optimal control $u_{opt}$ obtained by using adaptive-critic techniques) of the form

$$\dot{x}_{1d} = f_1(x_{1d}) + g_1(x_{1d})x_{2d}$$

(7)

$$\dot{x}_{2d} = f_2(x_{1d}, x_{2d}) + g_2(x_{1d}, x_{2d})u_{opt}$$

(8)

where $x_{1d} \in \mathbb{R}^{n_1}$, $x_{2d} \in \mathbb{R}^{n_2}$, $u_{opt} \in \mathbb{R}^{n_2}$ and $g_1 \in \mathbb{R}^{n_1 \times n_2}$, $g_2 \in \mathbb{R}^{n_2 \times n_2}$, $g_2^{-1}$ exists.

Suppose the true system contains input uncertainties and is of the form:

$$\dot{x}_1 = f_1(x_1) + g_1(x_1)x_2$$

(9)

$$\dot{x}_2 = f_2(x_1, x_2) + g_2(x_1, x_2)(I + \Delta(x_1, x_2))u_{opt}$$

(10)
Fig 1  Training the critic network

Fig 2  Training the action network
where $\Delta(x_1, x_2)$, the input uncertainty can be a function of the states but is bounded and
\[ \|\Delta(x_1, x_2)\| \leq 1 - \epsilon_g \text{ with } 0 < \epsilon_g \leq 1. \]

In order to deal with the uncertainty and make the perturbed system behave like Eqs. (7)-(8) (as originally intended), an extra-control ($u_e$) is added to Eq. (10):

\[
\dot{x}_2 = f_2(x_1, x_2) + g_2(x_1, x_2)(I + \Delta(x_1, x_2))(u_{opt} + u_e)
\]

(11)

This extra control is mainly composed of the output from an online-line tuned neural network which will be discussed later. The main property of neural network needed for control purpose is the function approximation. Let $f(x)$ be a smooth function from $\mathbb{R}^n \rightarrow \mathbb{R}^m$. It can be shown that for some sufficient large number of neurons, there exist weights ($W$) and activate function ($\varphi(x)$) such that

\[
f(x) = W^T \varphi(x) + \epsilon(x)
\]

(12)

where $\epsilon(x)$ is the neural network functional approximation error. In fact, for some positive number $\epsilon_N$, one can find a neural network such that $\|\epsilon(x)\| \leq \epsilon_N$. For good approximations, $\varphi(x)$ should be a basis such as gaussian, logsigmoid and so on.

**3.2 EXTRA CONTROL ($u_e$) DESIGN**

The goal is to find an extra control that can handle stabilize the system and make it behave like the original closed loop design in the presence of uncertainties. To be specific, make $x_1$ and $x_2$ bounded around the desired optimal trajectories. Here, an online tuned neural network is used for this purpose. Note that although our approach is Lyapunov based as in [27] and [28] we differ in the uncertainty model description.
In Eq. (9), $\dot{x}_{id}$ is subtracted on both sides to yield

$$
\dot{e}_1 = f_1(x_1) + g_1(x_1)x_2 - \dot{x}_{id} \\
= f_1 + g_1\alpha_2 - \dot{x}_{id} + g_1(x_2 - x_{2d}) - \alpha_2 + g_1^T \left( \frac{\partial V_1}{\partial x_1} \right) - g_1 g_1^T \left( \frac{\partial V_1}{\partial x_1} \right) + g_1 x_{2d}
$$

(13)

where $e_1 = x_1 - x_{id}$ and $V_1$ is an associated Lyapunov function and $\alpha_2$ is a stabilizing control for

$$
\dot{e}_1 = f_1(x_1) + g_1(x_1)u - \dot{x}_{id}
$$

(14)

For simplicity, the argument $x_1$ is omitted in the expressions of $f_1$ and $g_1$ in Eq. (13).

$$
\dot{V}_1 = (\frac{\partial V_1}{\partial t}) + (\frac{\partial V_1}{\partial e_1})^T (f_1 + g_1\alpha_2 - \dot{x}_{id}) + (\frac{\partial V_1}{\partial e_1})^T g_1 z -
$$

$$
(\frac{\partial V_1}{\partial e_1})^T g_1 g_1^T (\frac{\partial V_1}{\partial e_1}) + (\frac{\partial V_1}{\partial e_1})^T g_1 x_{2d}
$$

$$
\leq -\gamma_3 (|e_1|) + Az - AA^T + Ax_{2d}
$$

$$
\leq -\gamma_3 (|e_1|) - \left( \frac{1}{2} \|A\|^2 - \|z\|^2 \right) - \left( \frac{1}{2} \|A\|^2 - \|x_{2d}\|^2 \right) - \frac{1}{2} \|A\|^2 + \|z\|^2 + \|x_{2d}\|^2
$$

(15)

where $A = (\frac{\partial V_1}{\partial e_1})^T g_1$, $z = x_2 - x_{2d} - \alpha_2 + g_1^T (\frac{\partial V_1}{\partial e_1})$

and $(\frac{\partial V_1}{\partial t}) + (\frac{\partial V_1}{\partial e_1})(f_1 + g_1\alpha_2 - \dot{x}_{id}) \leq -\gamma_3 (|e_1|)$.

From Eq. (15), if $z$ is bounded, so are $V_1$ and $e_1$. Consider the derivative of $z$

$$
\dot{z} = \dot{x}_2 - \dot{x}_{2d} - \dot{\alpha}_2 + (g_1^T (\frac{\partial V_1}{\partial e_1})) = \dot{x}_2 - \dot{x}_{2d} - \dot{\alpha}_2 + G(x_1, x_{id})
$$

(16)

where $G(x_1, x_{id}) = (g_1^T (\frac{\partial V_1}{\partial e_1})) = d (g_1^T (\frac{\partial V_1}{\partial e_1})) / dt$

Insert Eq. (11) into Eq. (16) to get

$$
\dot{z} = f_2 + g_2 (I + \Delta)(u_{op} + u_e) - \dot{x}_{2d} - \dot{\alpha}_2 + G
$$

(17)

By choosing

$$
u_e = -g_2^{-1} (K_e e_2 + \tilde{f})
$$

(18)

where $e_2 = x_2 - x_{2d}$ and $\tilde{f}$ is the output of a neural network with $x_1, x_2, x_{id}, x_{2d}, e_1$ and $e_2$ as inputs. The role of $-K_e e_2$ is to provide stability that will help with initial convergence.

By substituting Eq. (18) into Eq. (17) we get
\[ z = -g_2(I + \Delta)g_2^{-1}K_z z + f_2 + g_2(I + \Delta)u_{opt} + G - \dot{\alpha}_2 - \dot{x}_{2d} + g_2(I + \Delta)g_2^{-1}K_z (A^T - \alpha_2) - g_2(I + \Delta)g_2^{-1}f \]  

(19)

By choosing a proper weight-update rule of this corrective neural network, the extra control \( u_e \) in Eq. (18) can make \( z \) bounded. This implies that \( e_1 \) and \( e_2 \) are bounded. Note that the errors do not go to zero but are bounded. We call this practical stability. The problem is how to find a weight-update rule to effect this sequence. We pick the structure of the neural network for \( u_e \) with two layers with weights of the first layer are fixed and weights of the output layer can be changed online.

Assume there exists ideal weights, such that

\[
f_2 + g_2(I + \Delta)u_{opt} + g_2(I + \Delta)g_2^{-1}K_z (A^T - \alpha_2) + AA^T z / \|z\|^2 + \alpha_2^T \alpha_2 z / \|z\|^2 + G - \dot{\alpha}_2 - \dot{x}_{2d} = W^T \varphi(\text{net}) + \varepsilon(x_1, x_2)
\]

(20)

with \( \|\varepsilon(x_1, x_2)\| < \mathcal{E}_N \) and \( \|W\|_F < \mathcal{W}_N \). Where \( \|\cdot\|_F \) is Frobenius norm and \( \|A\|^2_F = tr(A^T A) \). One of its properties is \( tr(A^T B) \leq \|A\|_F \|B\|_F \). For vectors, Frobenius norm is the same as 2-norm.

Let \( \tilde{W}^T \) denote real weights of the neural networks, that is

\[
\tilde{f} = \tilde{W}^T \varphi(\text{net})
\]

(21)

With Eq. (20) and Eq. (21), we can rewrite Eq. (19) as

\[
\dot{z} = -g_2(I + \Delta)g_2^{-1}K_z z - AA^T z / \|z\|^2 - \alpha_2^T \alpha_2 z / \|z\|^2 + \tilde{W}^T \varphi(\text{net}) - g_2\Delta g_2^{-1}(W - \tilde{W})^T \varphi(\text{net}) + \varepsilon
\]

(22)

where \( \tilde{W} = W - \tilde{W} \).

Claim 1: Updating rule \( \dot{\tilde{W}} = \varphi(\text{net})e^T - \gamma\tilde{W} \), where \( \gamma \) is the learning rate, achieves practical stability.

Proof: Choose a Lyapunov function as

\[
L = \frac{1}{2} z^T z + \frac{1}{2} tr(\tilde{W}^T \tilde{W})
\]

(23)
The rate of change of the Lyapunov function is given by

\[
\dot{L} = z^T \dot{z} + tr(W^T \dot{W})
\]

\[
= -z^T g_2 (I + \Delta) g_2^{-1} K_z z - z^T A A^T z r + z^T \alpha_2 z I ||z||^2 + z^T \varepsilon + z^T W^T \varphi - z^T g_2 \Delta g_2^{-1} (W - \tilde{W})^T \varphi + tr(-\tilde{W} T \varphi z^T) + tr(\tilde{W} T \varphi A) - tr(\tilde{W} T \varphi z^T)\tag{24}
\]

\[
\leq -\lambda_{min}(K_z) \varepsilon_{g} \|z\|^2 - \|A\|^2 - \|\alpha_2\|^2 + \|e_N + N^{1/2} (1 - \varepsilon_{g})\|_F W_N + \gamma \|\tilde{W}\|_F W_N - \gamma \|\tilde{W}\|^2_F + N^{1/2} (1 - \varepsilon_{g}) \|\tilde{W}\|_F + N^{1/2} \|\tilde{W}\|_F^2 + N^{1/2} \|\tilde{W}\|_F \|\alpha_2\|\tag{25}
\]

where \( N \) is the number of neurons at the output layer.

\[
\leq \left(-\lambda_{min}(K_z) \varepsilon_{g} \frac{N (1 - \varepsilon_{g})^2}{\gamma} \right) \|z\|^2 + \|e_N - \gamma \left(\frac{1}{2}\|\tilde{W}\|_F - W_N\right)^2 - \gamma \left(\frac{1}{2}\|\tilde{W}\|_F - \frac{N^{1/2}}{2} (1 - \varepsilon_{g})\right)^2 - \frac{N^{1/2}}{2} (1 - \varepsilon_{g}) \|e_N\|^2 \right)
\]

\[
- \left(\frac{N^{1/2}}{2} \|\tilde{W}\|_F - \|A\|^2 \right) - \left(\frac{N^{1/2}}{2} \|\tilde{W}\|_F - \|\alpha_2\|^2 \right) + W_N \|z\|_N^{1/2} (1 - \varepsilon_{g}) + \gamma W_N^2 - \frac{1}{2} (\gamma - N) \|\tilde{W}\|^2_F \right) \tag{26}
\]

If

\[
\|z\|_2 \geq \frac{e_N + W_N N^{1/2} (1 - \varepsilon_{g})}{\lambda_{min}(K_z) \varepsilon_{g} - N (1 - \varepsilon_{g})^2 / \gamma} + \frac{\gamma^{1/2} W_N}{(\lambda_{min}(K_z) \varepsilon_{g} - N (1 - \varepsilon_{g})^2 / \gamma)^{1/2}}, \quad \lambda_{min}(K_z) \varepsilon_{g} > \frac{N (1 - \varepsilon_{g})^2}{\gamma}
\]

and \( \gamma > N \),

then \( \dot{L} \leq 0 \), that is, \( L \) is bounded and then \( z \) is bounded. That makes \( e_1 \) and \( e_2 \) bounded.

We have shown that with an extra control computed online and applied to the plant along with the previously obtained control, the trajectory of the plant with input uncertainty will be close to the optimally designed path.

4. NUMERICAL RESULTS

In order to demonstrate the usefulness and the potential of the theory of robust adaptive critic based controllers, we present a numerical example. The equations of motion of a simplified nonlinear autopilot are presented below. These equations represent the missile’s short-period dynamics. The objective of this problem is to track reference commands in angle of attack.

\[
\dot{\alpha} = q - (K_1 + K_2 \alpha^2) \alpha
\]
\[ \dot{q} = (C_1 + C_2\alpha^2)\alpha + (C_3 + C_4\alpha^2)\bar{u} \] 
\[ \hat{\bar{u}} = -1/\tau(\bar{u} - u) \]

where \( \tau = 0.0001 \), \( \kappa_1 = 1.02 \), \( \kappa_2 = 1.28 \), \( c_1 = -57.17 \), \( c_2 = -322.16 \), \( c_3 = -70.12 \), and \( c_4 = -360.27 \).

Note that \( \alpha \) represents the angle of attack, \( \dot{q} \) represents the pitch rate, and \( u \) represents the fin position which is the control in this problem. The time-constant of the actuator is presented by \( \tau \). In this problem, we pick the variations or uncertainties in the values of the time-constant to reflect the input dynamics uncertainties. Reference angle of attack changes from 11.5 to -11.5 at 4s (800\textsuperscript{th} step).

At first, we solve this nonlinear control problem using adaptive critic technique with a quadratic cost function given by

\[ J = \sum_{i=0}^{\infty} ((x_i - x_{id})^T Q(x_i - x_{id}) + (u_i - u_{id})^T R(u_i - u_{id})) \]  

where \( Q = \text{diag}\left[\sqrt{\alpha_{\text{max}}}^2, \sqrt{q_{\text{max}}}^2, \sqrt{\bar{u}_{\text{max}}}^2\right], \) \( R = 1/u_{\text{max}}^2 \) with \( \alpha_{\text{max}} = 10/57.3, \) \( q_{\text{max}} = 0.5, \) \( u_{\text{max}} = 10/57.3, \) \( \bar{u}_{\text{max}} = 10/57.3, \) \( x = [\alpha \quad q \quad \bar{u}]^T, \) and sample frequency is 200Hz.

Two neural networks are used for critic (N3-8-8-3) and action (N3-6-6-1) neural networks. The training process is converged after 30 steps back and forth training. The trained action neural network will be used to provide optimal control.

In the second phase we change the time-constant by different values and investigate the robustness of the neurocontrollers. The actuator time constant is considered as the input uncertainty. Histories of the angle of attack (\( \alpha \)), control (\( u \)) and extra-control (\( u_e \)) are presented in Figures 3-8. The nonlinear adaptive critic based controller was designed with a \( \tau \) of 0.0001 (represented in the plot by nominal), however, it tracks very well with virtually no
difference when \( \tau \) changes from 0.0001 to 0.05 as in Figure 3. The control histories for these cases are presented in Figure 4. As a comparison, the performance of the inversion-based neurocontroller in [28], degrades very quickly when the design time constant is varied. Our hypothesis is that the inherent robustness of the adaptive critic based neurocontroller is due to the fact that they are based on optimality. Equivalent theory for linear systems is known. For a linear multivariable system, it has been shown that the controller synthesized with a quadratic cost function exhibits infinite gain margin and 60 degrees phase margin. In order to demonstrate the corresponding robustness property for linear systems, we obtained a linear model of the system (nonlinear parts are omitted). We found the optimal gains with a quadratic cost function and a \( \tau \) equal to 0.0001. Then, we changed the time constant from 0.0001 to 0.05 and obtained the performance of the linear optimal tracker. It is clear from simulation results in Figures 5 and 6 that the linear optimal tracker is robust. It appears that this robustness property is carried over to the class of nonlinear optimal controller also, although measures of robustness for nonlinear systems are hard to define. However, when the time-constant is raised to 0.5, there is need for extra-control to improve tracking can be observed in Figure 7. The histories of the control and extra-control are presented in Figure 8. This demonstrates the utility of the new theory developed in the last section to extend the region of robustness to input uncertainties.

5. CONCLUSIONS

Conditions for robustness for the adaptive critic based neurocontrollers have been derived with respect to unmodeled input dynamics. Simulation results show that the adaptive critic based controllers have excellent stability margins and can stabilize nonlinear system even with large uncertainties. This work was supported by a U.S. Army Space and Missile Defense Command Grant No. DAS60-99-C-0069 and a National Science Foundation Grant No. ECS-9976588.
REFERENCES


