JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 28, No. 3, May–June 2005

Robust Neurocontrollers for Systems with Model Uncertainties: A Helicopter Application

Zhongwu Huang* and S. N. Balakrishnan†
University of Missouri–Rolla, Rolla, Missouri 65401

A two-neural network approach to solving nonlinear optimal control problems is described. 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, a Lyapunov theory-based analysis for robust stability of the system under model uncertainties is developed and an extra control is developed. This extra control added with the basic control effort from the adaptive critic method guarantees good system performance and stablity under model uncertainties. This approach is demonstrated through a helicopter problem.

I. Introduction

OUTSIDE of dynamic programming,[1] currently there is no unified mathematical formalism under which a controller can be designed for nonlinear systems. Techniques such as feedback linearization have been used for a few nonlinear problems. It 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, and one can design the controller for the resulting linear system with some standard method. Then the control and state can be transformed back to original coordinates via an inverse transformation, and the net result will be a nonlinear controller. The conditions for the validity of such transformations have been studied in detail.[2–4] A particular case of feedback linearizing control, called dynamic inversion, has been studied in great detail for application to a supermaneuverable aircraft.[5–7] However, dynamic inversion is sensitive to modeling errors. In addition to feedback linearization, there are other methods for nonlinear control design. Chen and Narendra[8] proposed a design method by the nonlinear autoregressive moving average (NARMA). Sontag[9] and Sontag and Wang[10] developed an approach called input-to-state stabilization There are also methods based on H^∞ control[11–13] and formulations such as backstepping based on Lyapunov theorems (see Ref. 14).

A unique universal method to deal with control of any system, linear or nonlinear, is the optimal control theory. Problems of optimization of functions or functionals and optimal control of linear or nonlinear dynamic systems can be solved through direct or indirect methods.[1] The difficulty with optimal control of nonlinear systems is having to deal with Lagrange's multipliers (called costates in some studies). It is hard to have an intuitive idea about the magnitudes of the costates. 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, a neural network formulation is used that handles the Largrange multipliers (see Refs. 15–17) as outputs of a neural network.

Recently, the field of neural networks has become popular as a framework for formulating control problems. One of the earliest papers on neural adaptive control was published by Narendra and Parthasarathy.[18] An adaptive neurocontroller with guaranteed stability was developed by Polycarpou and Ioannou[19] under certain assumptions. Further research in neural adaptive control was presented by Sanner and Slotine[20] and Lewis et al.[21] Kim and Calise,[22] Leitner et al.,[23] and McFarland et al.[24] used a single-layer neural network for adaptive control of feedback linearized aircraft model. Kim and Calise[22] developed an architecture in which neural networks approximately cancel residual nonlinearities remaining after feedback linearization. McFarland et al.[24] extended this technique to a tracking problem with input uncertainty. They showed how an on-line updated neural network can stabilize an autopilot where input uncertainty exists. A similar technique is used to handle the approximate error in the dynamic inversion process.[23] Recently, a dynamic inversion-based neural adaptive control method, termed pseudocontrol hedging, was presented by Johnson et al.[25] to deal with the input saturation and input rate saturation. However, the adaptive critic design is more complex than the neurocontrol methods used in many studies. Adaptive critic design has its origin in reinforcement learning, for example, see Barto et al.[26] In this paper, however, it is used as a computational tool to solve optimal control problems based on approximate dynamic programming. This formulation obtains a nonlinear feedback control using feedforward neural networks. Balakrishnan and Biega[15] have shown the usefulness of this architecture for infinite-time linear problems.

Note that the adaptive critic solutions are based on an underlying system model, not the system itself. In many cases, system dynamics cannot be modeled perfectly. In this paper, the adaptive critic method is used to generate the optimal control law for the system model and an extra control is introduced through a Lyapunov analysis to deal with the uncertainty in the real system. The use of extra control with the adaptive critic method is similar to the use of adaptive control in dynamic inversion techniques, that is, using another control to handle the uncertainty. In this study, however, the real system is driven to be close to the performance of the optimal behavior of the system model. We present a fairly general adaptive critic formulation in Sec. II, and theory related to the robustness analysis and extra control design for systems affine in control are presented in Sec. III. Numerical results relating to a helicopter problem are presented in Sec. IV. Conclusions are drawn in Sec. V.

Received 11 April 2002; accepted for publication 12 February 2004.

*Ph.D. Student, Department of Mechanical and Aerospace Engineering and Engineering Mechanics; zhuang@umr.edu.

†Professor, Department of Mechanical and Aerospace Engineering and Engineering Mechanics; bala@umr.edu. Associate Fellow AIAA.

II. Problem Formulation and Solution Development

A. Dynamic Programming

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

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

In Eq. (1), the state at time step k is given by $\boldsymbol{x}(k) \in R^n$ and the control by $\boldsymbol{u}(k) \in R^m$. $J[\boldsymbol{x}(k)]$ represents the minimum cost associated with going from step k to the final step. $U[\boldsymbol{x}(k), \boldsymbol{u}(k)]$ is the utility function, which is the cost to go from k to $k+1$ using control $\boldsymbol{u}(k)$, and $J[\boldsymbol{x}(k+1)]$ is the minimum cost associated with going from step $k+1$ to the final step.

Define costate $\boldsymbol{\lambda}[\boldsymbol{x}(k)] \in R^n$ as

$$\boldsymbol{\lambda}[\boldsymbol{x}(k)] = \frac{\partial J[\boldsymbol{x}(k)]}{\partial \boldsymbol{x}(k)} \tag{2}$$

Equation (2) can also be written in a backward recurrent form as

$$\boldsymbol{\lambda}[\boldsymbol{x}(k)] = \frac{\partial U[\boldsymbol{x}(k), \boldsymbol{u}(k)]}{\partial \boldsymbol{x}(k)} + \frac{\partial \boldsymbol{u}(k)}{\partial \boldsymbol{x}(k)}\frac{\partial U[\boldsymbol{x}(k), \boldsymbol{u}(k)]}{\partial \boldsymbol{u}(k)} + \left[\frac{\partial \boldsymbol{x}(k+1)}{\partial \boldsymbol{x}(k)} + \frac{\partial \boldsymbol{u}(k)}{\partial \boldsymbol{x}(k)}\frac{\partial \boldsymbol{x}(k+1)}{\partial \boldsymbol{u}(k)}\right]^T \boldsymbol{\lambda}[\boldsymbol{x}(k+1)] \tag{3}$$

where we define

$$\frac{\partial \boldsymbol{Y}}{\partial \boldsymbol{X}} = \begin{bmatrix} \frac{\partial y_1}{\partial \boldsymbol{x}_1} & \cdots & \frac{\partial y_m}{\partial \boldsymbol{x}_1} \\ \cdots & \cdots & \cdots \\ \frac{\partial y_1}{\partial \boldsymbol{x}_n} & \cdots & \frac{\partial y_m}{\partial \boldsymbol{x}_n} \end{bmatrix}_{n \times m}$$

with $\boldsymbol{Y} = [y_1 \ \cdots \ y_m]_{m \times 1}$ and $\boldsymbol{X} = [\boldsymbol{x}_1 \ \cdots \ \boldsymbol{x}_n]_{n \times 1}$. From Eq. (1), Bellman's optimality equations are given by

$$\frac{\partial J[\boldsymbol{x}(k)]}{\partial \boldsymbol{u}(k)} = \left[\frac{\partial \boldsymbol{x}(k+1)}{\partial \boldsymbol{u}(k)}\right]^T \boldsymbol{\lambda}[\boldsymbol{x}(k+1)] + \frac{\partial U[\boldsymbol{x}(k), \boldsymbol{u}(k)]}{\partial \boldsymbol{u}(k)} = 0 \tag{4}$$

Equations (3) and (4) should be solved in conjunction with the underlying dynamic model for the optimal policy.

In this paper, the utility function $U[\boldsymbol{x}(k), \boldsymbol{u}(k)]$ is the popularly used quadratic function

$$U[\boldsymbol{x}(k), \boldsymbol{u}(k)] = \tfrac{1}{2}[\boldsymbol{x}^T(k)\boldsymbol{Q}\boldsymbol{x}(k) + \boldsymbol{u}^T(k)\boldsymbol{R}\boldsymbol{u}(k)] \tag{5}$$

The underlying system model is given by

$$\boldsymbol{x}(k+1) = \boldsymbol{f}[\boldsymbol{x}(k), \boldsymbol{u}(k)] \tag{6}$$

where $\boldsymbol{f}(.)$ can be either linear or nonlinear.

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

B. Adaptive Critic

Adaptive critic methodology has been proposed as an optimization technique combining concepts of reinforcement learning and dynamic programming. The goal is to find the control that minimizes the cost in Eq. (1) by solving Eqs. (3) and (4) with the use of Eq. (6) and the known initial states. To accomplish this task, we use two networks as in Figs. 1 and 2. One network, called *action*, models the control $\boldsymbol{u}(k)$ for which the inputs are the current states $\boldsymbol{x}(k)$. The other, called *critic*, models the costate $\boldsymbol{\lambda}(k)$, for which the inputs are also $\boldsymbol{x}(k)$. To train the critic network as in Fig. 1, first, $\boldsymbol{x}(k)$ is randomized and input to the action network to output $\boldsymbol{u}(k)$. The system model in Eq. (6) is then used to find $\boldsymbol{x}(k+1)$. The derivatives $\partial \boldsymbol{x}(k+1)/\partial \boldsymbol{u}(k)$, $\partial \boldsymbol{x}(k+1)/\partial \boldsymbol{x}(k)$, $\partial U(k)/\partial \boldsymbol{u}(k)$, and $\partial U(k)/\partial \boldsymbol{x}(k)$ can be calculated because $\boldsymbol{x}(k)$ and $\boldsymbol{u}(k)$ are known. Now, a randomized critic network is considered, and $\boldsymbol{\lambda}(k)$ and $\boldsymbol{\lambda}(k+1)$ are calculated corresponding to $\boldsymbol{x}(k)$ and $\boldsymbol{x}(k+1)$. With $\boldsymbol{\lambda}(k+1)$, the target $\boldsymbol{\lambda}(k)$, denoted $\boldsymbol{\lambda}^*(k)$, can be calculated by using Eq. (3). The difference between $\boldsymbol{\lambda}^*(k)$ and $\boldsymbol{\lambda}(k)$ 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 Fig. 2. This is done by finding $\boldsymbol{u}(k)$ for random $\boldsymbol{x}(k)$ and correcting them through the use of the model equation in Eq. (6) to find $\boldsymbol{x}(k+1)$ and by the use of $\boldsymbol{x}(k+1)$ to find $\boldsymbol{\lambda}(k+1)$ from the critic network corresponding to $\boldsymbol{x}(k+1)$. By using $\boldsymbol{\lambda}(k+1)$ in Eq. (4), we can solve for the target $\boldsymbol{u}^*(k)$ and use it to correct the action network. This two-step procedure continues till a predetermined level of convergence is reached. Liu and Balakrishnan[27] have proved the convergence of this process for linear systems. It is still an open issue for nonlinear systems. What we can say here is that if the outputs of these two neural networks are mutually consistent, then Bellman's optimal condition is satisfied and the resulting control is optimal.

C. Numerical Results with Adaptive Critic

In this section, a helicopter tracking problem is solved through the adaptive critic method outlined in Secs. II.A and II.B. The following set of equations from Ref. 28 describes the vertical motions of an X-Cell 50 model helicopter mounted on a stand. The objective of this problem is to track optimally the reference height commands of the helicopter and the collective pitch angle of the rotor blades.

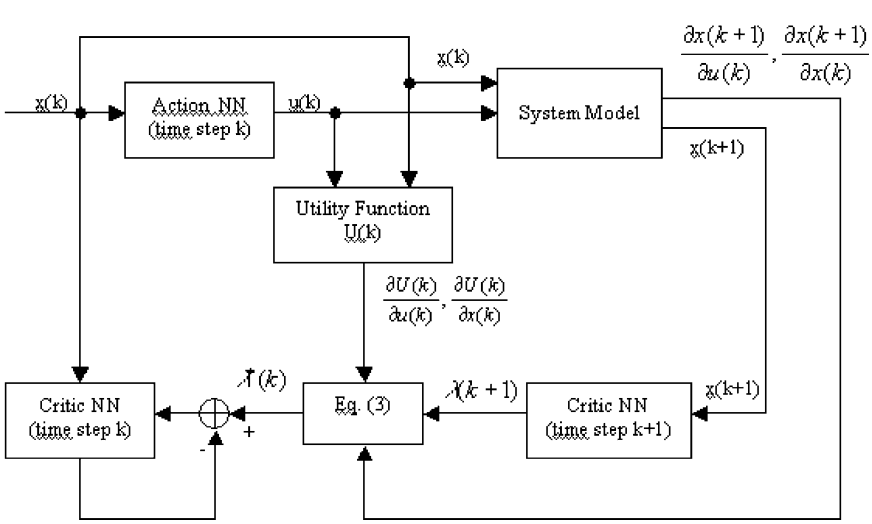

Fig. 1 Critic network training.

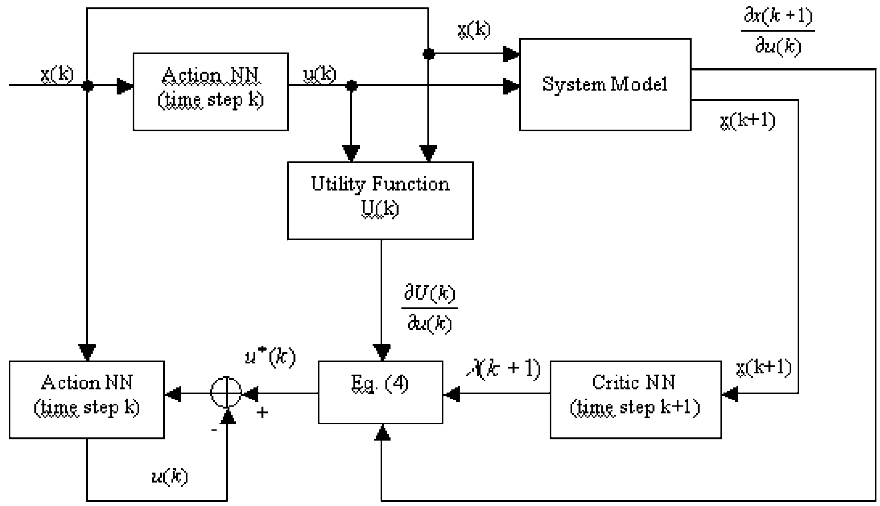

Fig. 2 Action network training.

This miniature helicopter is used as a test bed simply because of its availability in the published literature. Note that there is no specific application-oriented assumption in the development of adaptive critic formulation in this paper. Thus,

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3^2\left(a_1 + a_2 x_4 - \sqrt{a_3 + a_4 x_4}\right) + a_5 x_2 + a_6 x_2^2 + a_7$$

$$\dot{x}_3 = a_8 x_3 + a_{10} x_3^2 \sin x_4 + a_9 x_3^2 + a_{11} + u_1, \qquad \dot{x}_4 = x_5$$

$$\dot{x}_5 = a_{13} x_4 + a_{14} x_3^2 \sin x_4 + a_{15} x_5 + a_{12} + u_2 \tag{7}$$

where $a_1 = 5.31 \times 10^{-4}$, $a_2 = 1.5364 \times 10^{-2}$, $a_3 = 2.82 \times 10^{-7}$, $a_4 = 1.632 \times 10^{-5}$, $a_5 = a_6 = -0.1$, $a_7 = -17.66$, $a_8 = -0.7$, $a_9 = a_{10} = -0.0028$, $a_{11} = -13.92$, $a_{12} = 434.88$, $a_{13} = -800$, $a_{14} = -0.1$, and $a_{15} = -65$. Note that $\boldsymbol{x} = [x_1\ x_2\ x_3\ x_4\ x_5]$ and $\boldsymbol{u} = [u_1\ u_2]$. Here, x_1 is the height of the helicopter above the ground, x_3 is the rotational speed of the rotor blades, and x_4 is the collective pitch angle of the rotor blades. Also u_1 is the throttle control and u_2 the control from the collective servomechanisms. The desired heights of the helicopter and the collective pitch angle of the rotor blades are 1.25 m and 0.2 rad, respectively. On the basis of these, we solved for the rest of the steady-state values.

The adaptive critic technique is used to find a controller minimizing a quadratic cost function given by

$$J(k) = \sum_{i=k}^{\infty} \left\{ [(\boldsymbol{x}(i) - \boldsymbol{x}_r]^T Q[\boldsymbol{x}(i) - \boldsymbol{x}_r] + [\boldsymbol{u}(i) - \boldsymbol{u}_r]^T R[\boldsymbol{u}(i) - \boldsymbol{u}_r] \right\} \tag{8}$$

where $Q = \mathrm{diag}\lfloor 1/x_{1\max}^2\ 1/x_{2\max}^2\ 1/x_{3\max}^2\ 1/x_{4\max}^2\ 1/x_{5\max}^2 \rfloor$, $R = [1/u_{1\max}^2\ 1/u_{2\max}^2]$ with $\boldsymbol{x}_{\max} = [1.5\ 1\ 100\ 0.25\ 2]$, $\boldsymbol{u}_{\max} = [150\ \ 100]$, $\boldsymbol{x}_r$ is the desired final value of state, $\boldsymbol{u}_r$ is the final control corresponding to $\boldsymbol{x}_r$, which is [119.28; −70.97], and the sample frequency is 100 Hz ($\Delta T = 0.01$ s).

Equation (8) can also be written as

$$J(k) = J(k+1) + U(k) \tag{9}$$

where

$$U(k) = [\boldsymbol{x}(k) - \boldsymbol{x}_r]^T Q[\boldsymbol{x}(k) - \boldsymbol{x}_r] + [\boldsymbol{u}(k) - \boldsymbol{u}_r]^T R[\boldsymbol{u}(k) - \boldsymbol{u}_r] \tag{10}$$

Thus, $\partial U(k)/\partial \boldsymbol{u}(k)$, $\partial U(k)/\partial \boldsymbol{x}(k)$, $\partial \boldsymbol{x}(k+1)/\partial \boldsymbol{u}(k)$, and $\partial \boldsymbol{x}(k+1)/\partial \boldsymbol{x}(k)$ needed in the Eqs. (3) and (4) are

$$\frac{\partial U(k)}{\partial \boldsymbol{u}(k)} = R[\boldsymbol{u}(k) - \boldsymbol{u}_r], \qquad \frac{\partial U(k)}{\partial \boldsymbol{x}(k)} = Q[\boldsymbol{x}(k) - \boldsymbol{x}_r]$$

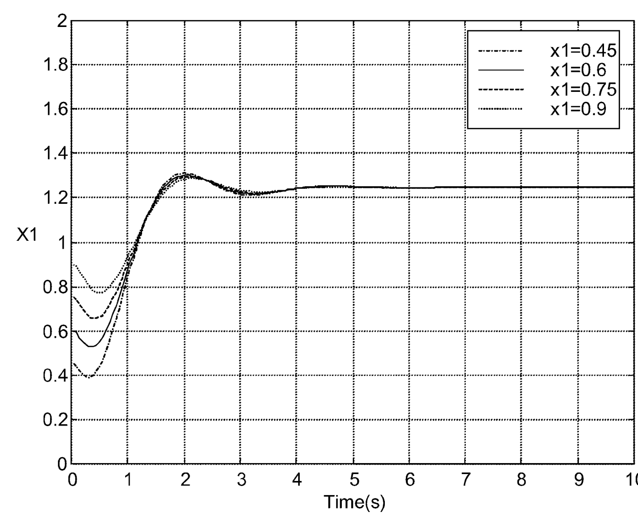

Fig. 3 Height of helicopter: different initial heights.

$$\frac{\partial \boldsymbol{x}(k+1)}{\partial \boldsymbol{u}(k)} = \Delta T \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\frac{\partial \boldsymbol{x}(k+1)}{\partial \boldsymbol{x}(k)} = \left[\frac{\partial \boldsymbol{x}_1(k+1)}{\partial \boldsymbol{x}(k)} \quad \cdots \quad \frac{\partial \boldsymbol{x}_5(k+1)}{\partial \boldsymbol{x}(k)} \right]_{5\times 5}$$

where $\partial \boldsymbol{x}_i(k+1)/\partial \boldsymbol{x}(k)$, $i = 1, \ldots, 5$, can be calculated from Eq. (7).

The architecture of critic neural networks is $N_{5-8-8-5}$, that is, five neurons corresponding to five states as inputs and five costates as outputs, and eight neurons for the first and second hidden layers. The architecture of action neural networks is $N_{5-8-8-2}$, that is, five neurons corresponding to five inputs states, two neurons corresponding to two outputs, and eight neurons for the first and second hidden layers. Our numerical results showed that this structure was adequate. The training process converged after 100 cycles of training. The trained action neural network will be used to provide optimal control for the system. Figures 3–6 show the simulation of the system corresponding to Eq. (7) with the optimal control obtained from the described adaptive critic method. The initial state of $\boldsymbol{x}_1$ is changed according to 0.45, 0.6, 0.75, and 0.9, whereas $\boldsymbol{x}_2$–$\boldsymbol{x}_5$ are unchanged and set to [0.1; 80; 0.1; 0.5]. Histories of the height of helicopter above the ground, $\boldsymbol{x}_1$; the collective pitch angle of the rotor blades, $\boldsymbol{x}_4$; and controls u_1 and u_2 are presented in Fig. 6. The optimal control from the adaptive critic technique works very well as shown in Fig. 6. The controls track accurately,

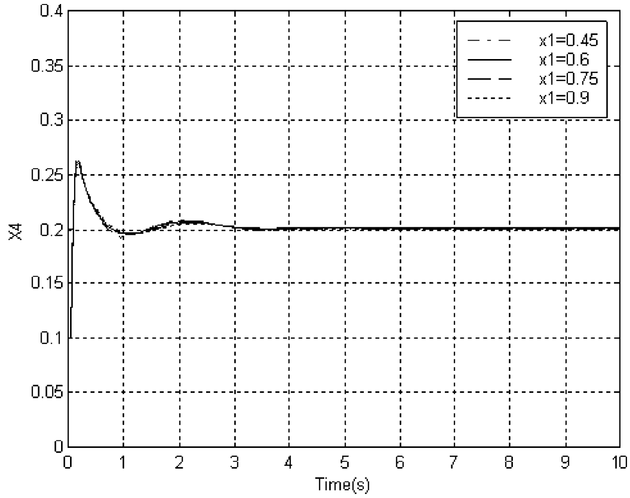

Fig. 4 Collective pitch angle: different initial heights.

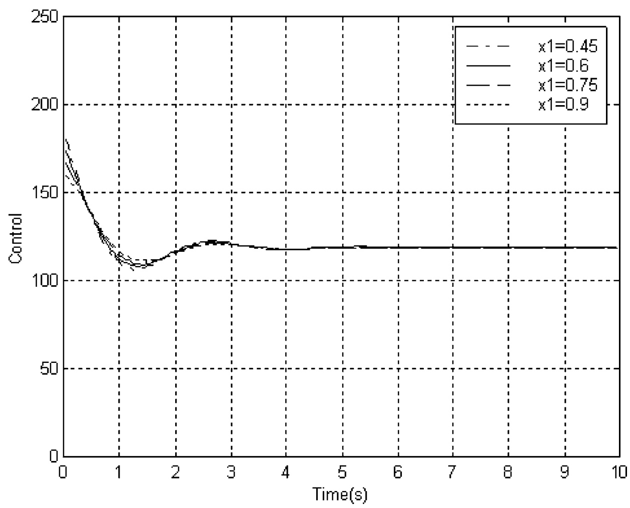

Fig. 5 Control from the throttle, u_1: different initial heights.

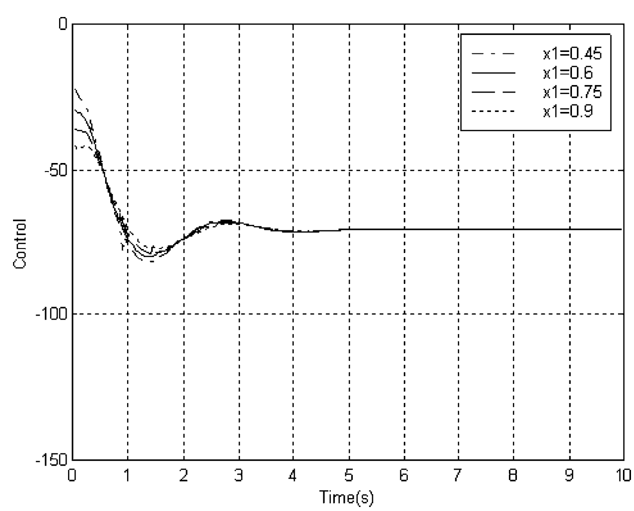

Fig. 6 Control from the collective servomechanism, u_2: different initial heights.

and they respond well to variable initial conditions. Note that the controller is synthesized off-line but is used as a feedback controller online.

However, the optimality of the adaptive critic technique is limited by the underlying system model. If the model has uncertainties, its performance may not be adequate or even stable with the optimal control. In the next section, a method using extra control is developed to help the system yield performance close to the optimal level while operating under model uncertainties; input uncertainty is not considered in this paper.

III. Robust Design

A. Problem Description

Consider a nominal nonlinear affine system (with optimal control u_{opt} obtained by using the adaptive critic technique):

$$\dot{\boldsymbol{x}}_{1d} = \boldsymbol{f}_1(\boldsymbol{x}_{1d}, \boldsymbol{x}_{2d}) \tag{11}$$

$$\dot{\boldsymbol{x}}_{2d} = \boldsymbol{f}_2(\boldsymbol{x}_{1d}, \boldsymbol{x}_{2d}) + g_2(\boldsymbol{x}_{1d}, \boldsymbol{x}_{2d})\boldsymbol{u}_{\text{opt}}(\boldsymbol{x}_{1d}, \boldsymbol{x}_{2d}) \tag{12}$$

where $\boldsymbol{x}_{1d} \in R^{n_1}$, $\boldsymbol{x}_{2d} \in R^{n_2}$, $\boldsymbol{u}_{\text{opt}} \in R^m$, and $g_2 \in R^{n_2 \times m}$. With model uncertainty,

$$\dot{x}_1 = \boldsymbol{f}_1(\boldsymbol{x}_1, \boldsymbol{x}_2) + \boldsymbol{d}_{11}(\boldsymbol{x}_1, \boldsymbol{x}_2) + \boldsymbol{d}_{12} \tag{13}$$

$$\dot{x}_2 = \boldsymbol{f}_2(\boldsymbol{x}_1, \boldsymbol{x}_2) + g_2(\boldsymbol{x}_1, \boldsymbol{x}_2)\boldsymbol{u}_{\text{opt}}(\boldsymbol{x}_1, \boldsymbol{x}_2) + \boldsymbol{d}_{21}(\boldsymbol{x}_1, \boldsymbol{x}_2) + \boldsymbol{d}_{22} \tag{14}$$

where $\boldsymbol{d}_{11}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ and $\boldsymbol{d}_{21}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ are model uncertainty. Here $\boldsymbol{d}_{12}$ and $\boldsymbol{d}_{22}$ are bounded noise with $\|\boldsymbol{d}_{11}\| < d_{1N}$ and $\|\boldsymbol{d}_{22}\| < d_{2N}$.

To deal with the uncertainty and make the perturbed system behave like Eqs. (11) and (12), an extra control $\boldsymbol{u}_e$ is added to Eq. (14):

$$\dot{x}_2 = \boldsymbol{f}_2(\boldsymbol{x}_1, \boldsymbol{x}_2) + g_2(\boldsymbol{x}_1, \boldsymbol{x}_2)(\boldsymbol{u}_{\text{opt}} + \boldsymbol{u}_e) + \boldsymbol{d}_{21}(\boldsymbol{x}_1, \boldsymbol{x}_2) + \boldsymbol{d}_{22} \tag{15}$$

The design of the extra-control will be discussed in the next section.

The key to using the neural networks for control is their function approximation property. Let $\boldsymbol{f}(\boldsymbol{x})$ be a smooth function from $\Re^n \to \Re^m$. It can be shown that within a compact set of state $\boldsymbol{x}$, for some sufficiently large number of neurons, there exist weights W and activation function $\varphi(\boldsymbol{x})$ such that[29]

$$\boldsymbol{f}(\boldsymbol{x}) = W^T\varphi(\boldsymbol{x}) + \varepsilon(\boldsymbol{x}) \tag{16}$$

where $\varepsilon(\boldsymbol{x})$ is the neural network functional approximation error. In fact, for some positive number ε_N, one can find a neural network (NN) such that $\|\varepsilon(\boldsymbol{x})\| \leq \varepsilon_N$.

B. Extra Control u_e Design

The goal is to find an extra control that can handle the model uncertainties of a system and help the system perform close to the nominal system behavior. To be specific, make $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ bounded around the desired trajectories. For many engineering problems, it is physically impossible to make the errors go to zero; however, it is possible (and acceptable) to keep the errors bounded. This is termed practical stability. A Lyapunov function-based method is used here to obtain this extra control; an online tuned neural network is used for the design of the extra control.

The error dynamics equations of the system are

$$\dot{e}_1 = \boldsymbol{f}_1(\boldsymbol{x}_1, \boldsymbol{x}_2) + \boldsymbol{d}_{11}(\boldsymbol{x}_1, \boldsymbol{x}_2) + \boldsymbol{d}_{12} - \dot{x}_{1d} \tag{17}$$

$$\dot{e}_2 = \boldsymbol{f}_2(\boldsymbol{x}_1, \boldsymbol{x}_2) + g_2(\boldsymbol{x}_1, \boldsymbol{x}_2)(\boldsymbol{u}_{\text{opt}} + \boldsymbol{u}_e) + \boldsymbol{d}_{21}(\boldsymbol{x}_1, \boldsymbol{x}_2) + \boldsymbol{d}_{22} - \dot{x}_{2d} \tag{18}$$

where $\boldsymbol{e}_1 = \boldsymbol{x}_1 - \boldsymbol{x}_{1d}$ and $\boldsymbol{e}_2 = \boldsymbol{x}_2 - \boldsymbol{x}_{2d}$. The objective here is to make $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ bounded. It will be shown later that this can be done by using an extra control as

$$\boldsymbol{u}_e = g_2^T\boldsymbol{e}_2 \Big/ \left\| g_2^T\boldsymbol{e}_2 \right\|^2 \left[-\boldsymbol{e}_2^T K_2\boldsymbol{e}_2 - \boldsymbol{e}_2^T\hat{\boldsymbol{F}}(\boldsymbol{x}_1, \boldsymbol{x}_2) - \boldsymbol{e}_1^T K_1\boldsymbol{e}_1 \right] \tag{19}$$

where $\hat{\boldsymbol{F}}(\boldsymbol{x}_1, \boldsymbol{x}_2)$ is the output of an NN with $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_{1d}$, $\boldsymbol{x}_{2d}$, $\boldsymbol{e}_1$, and $\boldsymbol{e}_2$ as inputs. The terms $-\boldsymbol{e}_1^T K_1\boldsymbol{e}_1$ and $-\boldsymbol{e}_2^T K_2\boldsymbol{e}_2$ are stabilizing terms, and they help with better initial convergence characteristics. As can be seen later in the Lyapunov function-based proof, proper values for k_1 and k_2 in Eq. (19) are needed to ensure that the state of real system will be in some compact region over which Eq. (16) is satisfied. Within that region, the NN in Eq. (19) will counteract those model uncertainties. It is not the same as the high gain control.[30,31]

By the choice of a proper weight-update rule for the NN, $\boldsymbol{u}_e$ in Eq. (19) can make the perturbed system practically stable, that is, $\boldsymbol{e}_1$ and $\boldsymbol{e}_2$ bounded. The problem is how to find such a weight-update rule. We pick the structure of the NN for $\boldsymbol{u}_e$ with three layers and weights of each layer to be changed online.

Assume there exists ideal weights such that

$$W_3^T\varphi_3\left\{W_2^T\varphi_2\left[W_1^T\varphi_1(P)\right]\right\}+\varepsilon(\boldsymbol{x}_1,\boldsymbol{x}_2) = \boldsymbol{f}_2(\boldsymbol{x}_1,\boldsymbol{x}_2)+g_2(\boldsymbol{x}_1,\boldsymbol{x}_2)\boldsymbol{u}_{\text{opt}}+\boldsymbol{d}_{21}(\boldsymbol{x}_1,\boldsymbol{x}_2)-\dot{x}_{2d} + \left(\boldsymbol{e}_2\boldsymbol{e}_1^T/\|\boldsymbol{e}_2\|^2\right)[\boldsymbol{f}_1(\boldsymbol{x}_1,\boldsymbol{x}_2)+\boldsymbol{d}_{11}(\boldsymbol{x}_1,\boldsymbol{x}_2)-\dot{x}_{1d}] \tag{20}$$

with $\|\varepsilon(\boldsymbol{x}_1,\boldsymbol{x}_2)\|<\varepsilon_N$, $\|W_1\|_F<W_{1N}$, $\|W_2\|_F<W_{2N}$, and $\|W_3\|_F<W_{3N}$, where $\|\cdot\|_F$ is the Frobenius norm and $\|A\|_F^2=\operatorname{tr}(A^TA)$. One of its properties is $\operatorname{tr}(A^TB)\le\|A\|_F\|B\|_F$. For vectors, the Frobenius norm is the same as the 2 norm.

Let $\widehat{W}_1$, $\widehat{W}_2$, and $\widehat{W}_3$, be the real weights of the NN and $\widehat{\varphi}_1$, $\widehat{\varphi}_2$, $\widehat{\varphi}_3$ the real outputs of each layer, that is,

$$\widehat{\boldsymbol{F}}(\boldsymbol{x}_1,\boldsymbol{x}_2)=\widehat{W}_3^T\widehat{\varphi}_3(\text{net}) \tag{21}$$

Insert Eq. (19) into Eq. (18); we get

$$\dot{e}_2=\boldsymbol{f}_2(\boldsymbol{x}_1,\boldsymbol{x}_2)+g_2(\boldsymbol{x}_1,\boldsymbol{x}_2)\boldsymbol{u}_{\text{opt}}+\boldsymbol{d}_{21}(\boldsymbol{x}_1,\boldsymbol{x}_2)+\boldsymbol{d}_{22}-\dot{x}_{2d} + \left(g_2g_2^T\boldsymbol{e}_2/\left\|g_2^T\boldsymbol{e}_2\right\|^2\right)\left[-\boldsymbol{e}_2^TK_2\boldsymbol{e}_2-\boldsymbol{e}_2^T\widehat{\boldsymbol{F}}(\boldsymbol{x}_1,\boldsymbol{x}_2)-\boldsymbol{e}_1^TK_1\boldsymbol{e}_1\right] \tag{22}$$

Claim 1: The following weight update rule achieves practical stability:

$$\dot{\widehat{W}}_1=-\gamma_1\widehat{\varphi}_1\left[\widehat{W}_1^T\widehat{\varphi}_1+B_{11}\boldsymbol{e}_1+B_{21}\boldsymbol{e}_2\right]^T$$

$$\dot{\widehat{W}}_2=-\gamma_2\widehat{\varphi}_2\left[\widehat{W}_2^T\widehat{\varphi}_2+B_{12}\boldsymbol{e}_1+B_{22}\boldsymbol{e}_2\right]^T$$

$$\dot{\widehat{W}}_3=\gamma_3\left(\widehat{\varphi}_3\boldsymbol{e}_2^T+\widehat{\varphi}_3(B_{13}\boldsymbol{e}_1)^T-\gamma\widehat{W}_3\right) \tag{23}$$

where B_{11}, B_{12}, B_{13}, B_{21}, and B_{22} are some coefficient weight matrices, γ_1, γ_2, and γ_3 are the learning rates, γ is a contraction rate, and $\|\widehat{\varphi}_1\|<\varphi_{1N}$, $\|\widehat{\varphi}_2\|<\varphi_{2N}$, and $\|\widehat{\varphi}_3\|<\varphi_{3N}$.

Proof: Choose a Lyapunov function as

$$L=\frac{1}{2}\boldsymbol{e}_1^T\boldsymbol{e}_1+\frac{1}{2}\boldsymbol{e}_2^T\boldsymbol{e}_2+\frac{1}{2}\sum_{i=1}^{3}\gamma_i^{-1}\operatorname{tr}\left(\tilde{W}_i^T\tilde{W}_i\right) \tag{24}$$

where $\tilde{W}_1=W_1-\widehat{W}_1$, $\tilde{W}_2=W_2-\widehat{W}_2$, and $\tilde{W}_3=W_3-\widehat{W}_3$. The derivative of L is

$$\dot{L}=\boldsymbol{e}_1^T\dot{e}_1+\boldsymbol{e}_2^T\dot{e}_2+\sum_{i=1}^{3}\gamma_i^{-1}\operatorname{tr}\left(\tilde{W}_i^T\dot{\tilde{W}}_i\right) \tag{25}$$

Insert Eqs. (20–23) into Eq. (25):

$$\le-\left(\lambda_{\min}(K_1)-\tfrac{1}{4}-\|B_{11}\|^2-\|B_{12}\|^2\right)\|\boldsymbol{e}_1\|^2-\left(\lambda_{\min}(K_2)-\tfrac{1}{4}-\|B_{21}\|^2-\|B_{22}\|^2\right)\|\boldsymbol{e}_2\|^2+\|\boldsymbol{e}_1\|d_{1N}+\|\boldsymbol{e}_2\|(d_{2N}+\varepsilon_N+2W_{3N}\varphi_{3N})-\left(3\gamma/4-\|B_{13}\|^2\varphi_{3N}^2\right)\|\tilde{W}_3\|_F^2+\gamma W_{3N}^2+\varphi_{1N}^2W_{1N}^2+\varphi_{2N}^2W_{2N}^2 \tag{26}$$

Pick $\lambda_{\min}(K_1)>\frac{1}{4}+\|B_{11}\|^2+\|B_{12}\|^2$, $\lambda_{\min}(K_2)>\frac{1}{4}+\|B_{21}\|^2+\|B_{22}\|^2$, and $\gamma>\frac{4}{3}\|B_{13}\|^2\varphi_{3N}^2$. Define $e=[\boldsymbol{e}_1^T,\boldsymbol{e}_2^T]^T$. Then,

$$\|\boldsymbol{e}_1\|_2^2+\|\boldsymbol{e}_2\|_2^2=\|e\|_2^2,\qquad\|\boldsymbol{e}_1\|_2^2\le\|e\|_2^2,\qquad\|\boldsymbol{e}_2\|_2^2\le\|e\|_2^2 \tag{27}$$

Let

$$a=\min\left\{\left[\lambda_{\min}(K_1)-\tfrac{1}{4}-\|B_{11}\|^2-\|B_{12}\|^2\right]\times\left[\lambda_{\min}(K_2)-\tfrac{1}{4}-\|B_{21}\|^2-\|B_{22}\|^2\right]\right\} \tag{28}$$

The Lyapunov derivative now satisfies the inequality

$$\begin{aligned}\dot{L}&\le-a\left(\|\boldsymbol{e}_1\|^2+\|\boldsymbol{e}_2\|^2\right)+\|\boldsymbol{e}_1\|d_{1N}+\|\boldsymbol{e}_2\|(d_{2N}+\varepsilon_N+2W_{3N}\varphi_{3N})\\&\quad-\left(3\gamma/4-\|B_{13}\|^2\varphi_{3N}^2\right)\|\tilde{W}_3\|_F^2+\gamma W_{3N}^2+\varphi_{1N}^2W_{1N}^2+\varphi_{2N}^2W_{2N}^2\\&\le-a\|e\|^2+\|e\|(d_{1N}+d_{2N}+\varepsilon_N+2W_{3N}\varphi_{3N})+\gamma W_{3N}^2\\&\quad+\varphi_{1N}^2W_{1N}^2+\varphi_{2N}^2W_{2N}^2\\&\triangleq-a\|e\|^2+b\|e\|+c\end{aligned} \tag{29}$$

$$b\triangleq(d_{1N}+d_{2N}+\varepsilon_N+2W_{3N}\varphi_{3N})\ge0$$

$$c\triangleq\gamma W_{3N}^2+\varphi_{1N}^2W_{1N}^2+\varphi_{2N}^2W_{2N}^2\ge0$$

When $\|e\|>\sigma$, where $\sigma=[-b+\sqrt{(b^2-4ac)}]/2a$,

$$\dot{L}<0 \tag{30}$$

Now, it is desired to define the region of validity of the whole Lyapunov analysis. The development in this paper follows Ref. 32. Suppose in domain $\mathbb{Q}$ of $\boldsymbol{P}$ (where $\boldsymbol{P}$ is the input vector to the NN) the ideal weight brings the term $W_3^T\varphi_3\{W_2^T\varphi_2[W_1^T\varphi_1(\boldsymbol{P})]\}$ to within a Δ neighborhood of the error ε_0, and Δ is bounded by

$$\Delta^*\equiv\sup_{\mathbb{Q}}\left|W_3^T\varphi_3\left\{W_2^T\varphi_2\left[W_1^T\varphi_1(\boldsymbol{P})\right]\right\}+\varepsilon(\boldsymbol{x}_1,\boldsymbol{x}_2)-\boldsymbol{F}(\boldsymbol{x})\right| \tag{31}$$

where

$$\boldsymbol{F}(\boldsymbol{x})=\boldsymbol{f}_2(\boldsymbol{x}_1,\boldsymbol{x}_2)+g_2(\boldsymbol{x}_1,\boldsymbol{x}_2)\boldsymbol{u}_{\text{opt}}+\boldsymbol{d}_{21}(\boldsymbol{x}_1,\boldsymbol{x}_2)-\dot{x}_{2d} + \left(\boldsymbol{e}_2\boldsymbol{e}_1^T/\|\boldsymbol{e}_2\|^2\right)[\boldsymbol{f}_1(\boldsymbol{x}_1,\boldsymbol{x}_2)+\boldsymbol{d}_{11}(\boldsymbol{x}_1,\boldsymbol{x}_2)-\dot{x}_{1d}] \tag{32}$$

The ideal neural network minimizes Δ^* over the domain $\mathbb{Q}$.

The matrix of weights are converted to vectors for later use:

$$\widehat{W}_i=\begin{bmatrix}w_{1,1}&\cdots&w_{1,n_i}\\\vdots&\ddots&\vdots\\w_{m_i,1}&\cdots&w_{m_i,n_i}\end{bmatrix}$$

$$\widehat{W}_i'=\left[w_{1,1}\ \cdots\ w_{1,n_i}\ w_{2,1}\ \cdots\ w_{2,n_i}\ \cdots\ w_{m_i,1}\ \cdots\ w_{m_i,n_i}\right]^T\qquad i=1,2,3$$

$$\|\widehat{W}_i\|_F=\|\widehat{W}_i'\|_2 \tag{33}$$

The extended state space in the Lyapunov analysis [refer to Eqs. (17), (18), (23), and (33)] is

$$\eta=\left[e^T\quad\widehat{W}_1'^T\quad\widehat{W}_2'^T\quad\widehat{W}_3'^T\right]^T \tag{34}$$

Define

$$B^r=\{\eta:|\eta|\le r\},\qquad\Omega_\alpha=\{\eta\in B^r:\eta^T\eta\le\alpha\}$$

$$\alpha=\min_{|\eta|=r}\eta^T\eta=r^2 \tag{35}$$

$$\Omega_\beta=\left\{\eta\in B^r:\|e\|_2\le\sigma\right\} \tag{36}$$

From Eq. (30), it can be seen that $\Omega_\alpha \subset B^r$ is an invariant set. If $\Omega_\beta \subset \Omega_\alpha$, then the minimum size of B^r can be quantified by

$$r > \sigma \tag{37}$$

Domain $\mathbb{Q}$ must be sufficiently large, so that $B^r \in \mathbb{Q}$.

This is sufficient to show, via the Lasalle theorem, that if $e(t_0) \in \Omega_\alpha$, then $e(t)$ and $\hat{W}_1$, $\hat{W}_2$, and $\hat{W}_3$ will remain bounded.

This analysis shows that with an extra control computed online and applied to the plant along with the control obtained earlier from the adaptive critic method the trajectory of the plant with uncertainty will be close to the optimally designed path.

IV. Numerical Results

To demonstrate the usefulness and the potential of the theory of robust adaptive critic-based controllers, uncertainties are added in the form of a bounded function of states to Eq. (7) and the robustness of the helicopter performance is investigated with use of the extra control $\boldsymbol{u}_e$. The perturbed system is described by

$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= x_3^2\left(a_1 + a_2 x_4 - \sqrt{a_3 + a_4 x_4}\right) + a_5 x_2 + a_6 x_2^2 + a_7 \\
\dot{x}_3 &= a_8 x_3 + a_{10} x_3^2 \sin x_4 + a_9 x_3^2 + a_{11} + u_1 + u_{e1} + c_1 \sin(x_3/50) \\
\dot{x}_4 &= x_5 \\
\dot{x}_5 &= a_{13} x_4 + a_{14} x_3^2 \sin x_4 + a_{15} x_5 + a_{12} + u_2 + u_{e2} + c_2 \sin(x_5)
\end{aligned} \tag{38}$$

where $c_1 = 60$ and $c_2 = 20$.

Here, the parameters for the extra control and the weight update rule are chosen as the following: Gain values are $K_1 = [0.5\ 0\ 0;\ 0\ 0.5\ 0;\ 0\ 0\ 0.5]$, $K_2 = [5\ 0;\ 0\ 5]$, $B_{11} = B_{12} = B_{13} = 0$, and $B_{21} = B_{22} = 5$. Learning rate values are $\gamma_1 = 0.1$, $\gamma_2 = 0.1$, $\gamma_3 = 0.5$, and $\gamma = 0.01$. Note that B_{11}, B_{12}, and B_{13} are set to 0 because their corresponding parts in the weight update rule are not needed in this particular problem. To use every input equally, the inputs $\boldsymbol{x}$, x_d, and e to the extra control's NN are normalized to $-\alpha$ and α with $(\{\alpha/[1+\exp(-\alpha_1 z)]\} - \{\alpha/[1+\exp(\alpha_1 z)]\})$, where z is $\boldsymbol{x}$, x_d, or e and $\alpha = 1$ and $\alpha_1 = [1;\ 1;\ 0.05;\ 5;\ 1]$. Here the α_1 components are chosen opposite the range of corresponding states of x_d.

Histories of the height of helicopter above the ground, $\boldsymbol{x}_1$, the collective pitch angle of the rotor blades, x_4, control $\boldsymbol{u}$, and extra control $\boldsymbol{u}_e$ are presented in Figs. 7–10 with the initial states of [0.5; 0.1; 70; 0.1; 0.5]. The nominal system in Figs. 7 and 8 corresponds to the system described by Eq. (7) with the optimal control from the adaptive critic method. The trajectories of this nominal

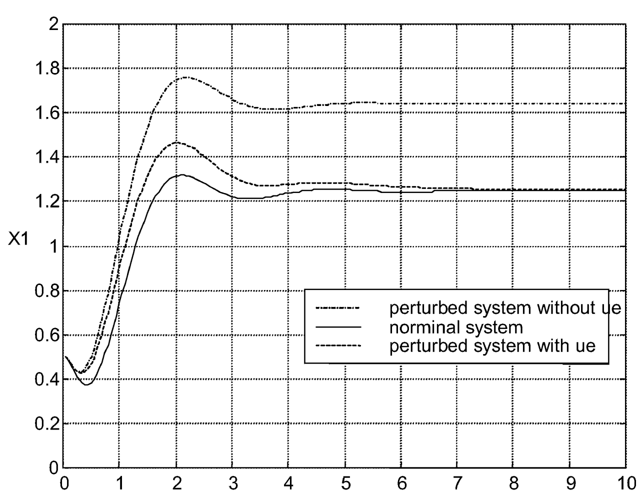

Fig. 7 Height of helicopter: model uncertainties included.

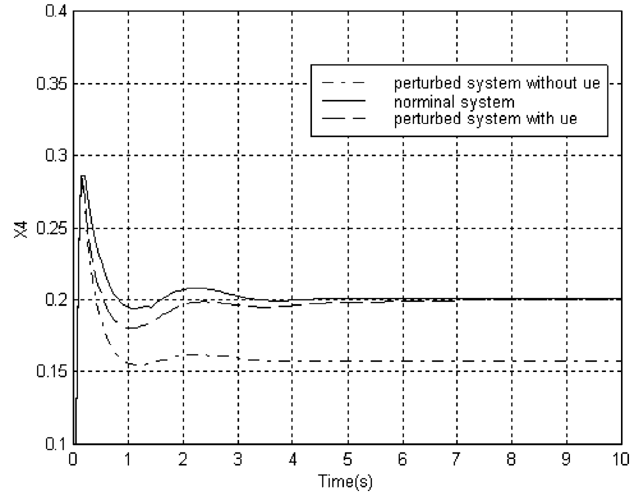

Fig. 8 Collective pitch angle: model uncertainties included.

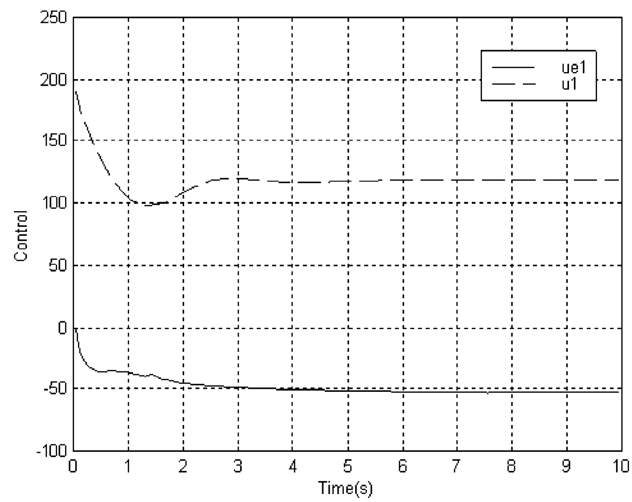

Fig. 9 Control from the throttle: model uncertainties included.

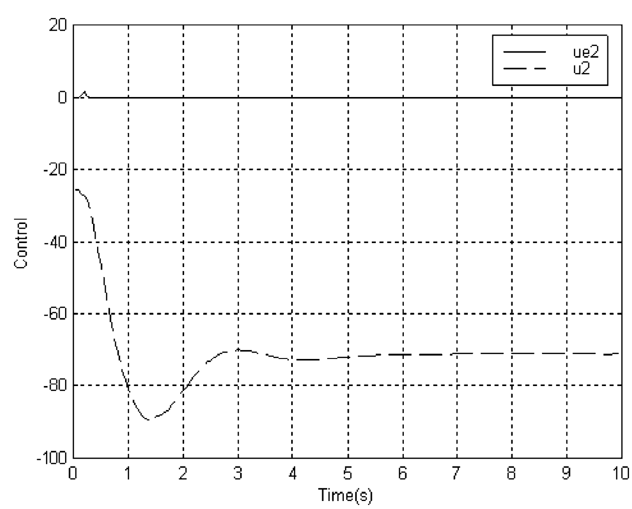

Fig. 10 Control from the collective servomechanism, u_2: model uncertainties included.

system are used as a reference in designing the extra control. The system described by Eq. (29) with the optimal control alone (without extra control) is denoted as perturbed system without $\boldsymbol{u}_e$. The simulations of this system are used for comparison with those of the system with extra control (represented in the plots by perturbed system with $\boldsymbol{u}_e$). Figures 7 and 8 show that in the presence of uncertainties the height of helicopter above the ground and the collective pitch angle of the rotor blades can reach the desired values of 1.25 m and 0.2 rad, respectively, by using the extra control. There would have been some errors in the final states if the extra control had not been applied. The histories of the optimal control and extra control from the throttle and the collective servomechanisms are presented in Figs. 9 and 10.

Then the initial state values are changed to study the sensitivity of the extra-control approach. Figures 11–14 correspond to the cases of 10% decrease on all of the initial values of $\boldsymbol{x}$, that is,

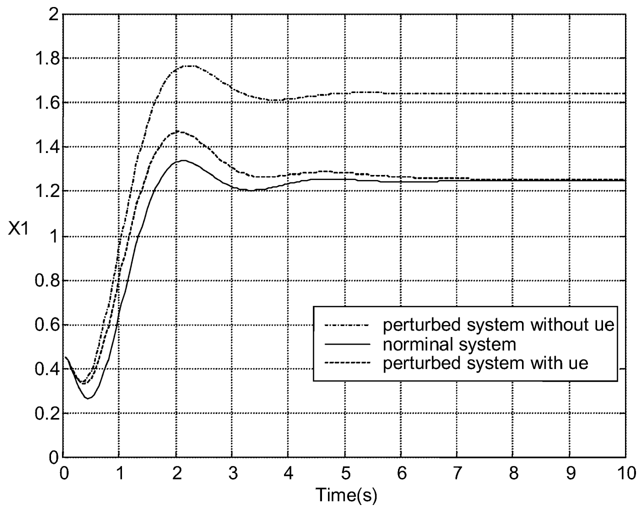

Fig. 11 Height of helicopter: 10% decrease in initial values.

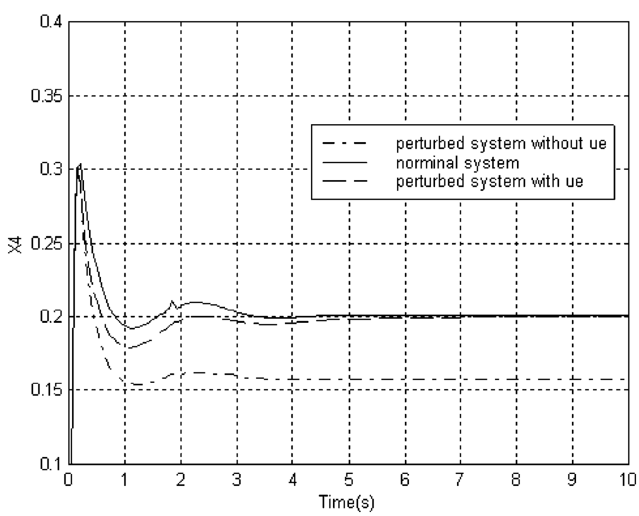

Fig. 12 Collective pitch angle: 10% decrease in initial values.

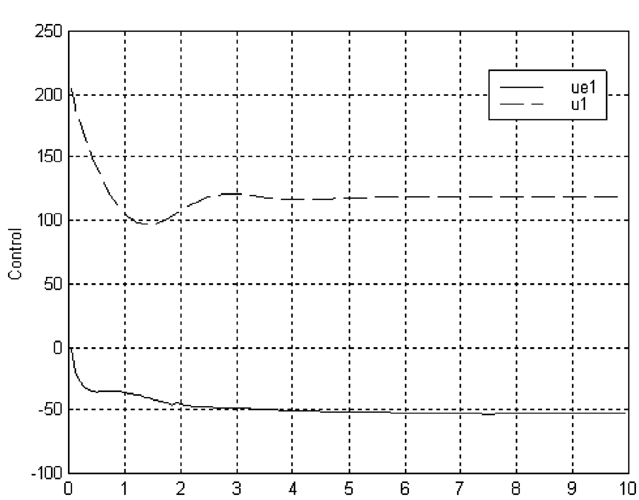

Fig. 13 Control from the throttle, u_1: 10% decrease in initial values.

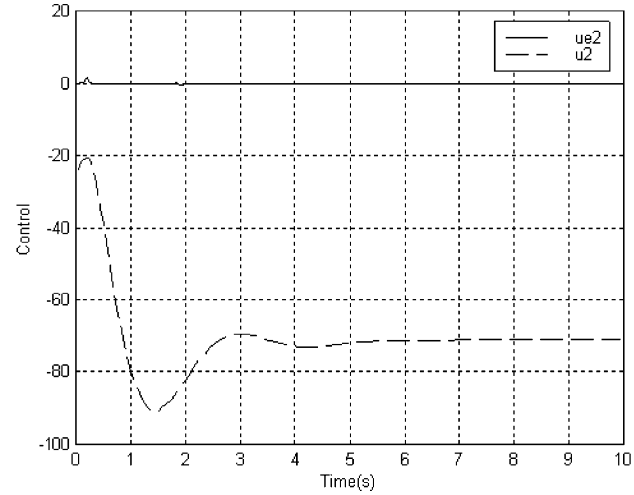

Fig. 14 Control from the collective servomechanism, u_2: 10% decrease in initial values.

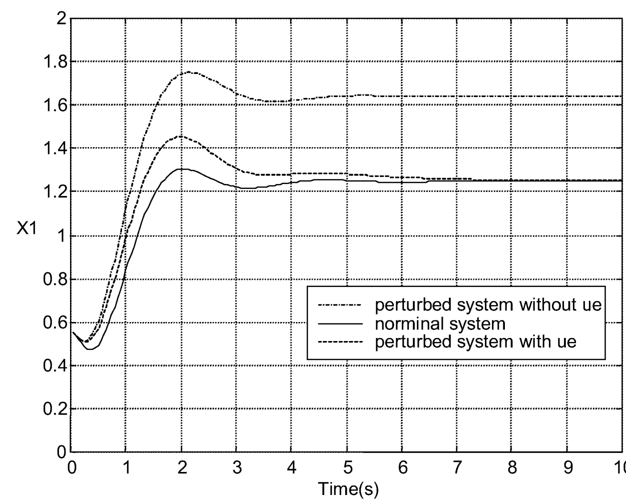

Fig. 15 Height of helicopter: 10% increase in initial values.

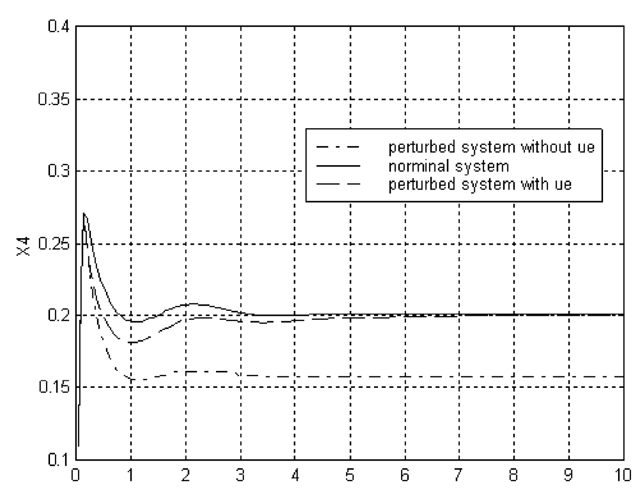

Fig. 16 Collective pitch angle: 10% increase in initial values.

[0.45; 0.09; 63; 0.09; 0.45]. Figures 11 and 12 show the histories of the height of helicopter and the collective pitch angle of the rotor blades. The histories of the optimal control and extra control from the throttle and the collective servomechanisms are presented in Figs. 13 and 14. Figures 15–18 correspond to the cases of 10% increase on all of the initial values of $\boldsymbol{x}$, that is, [0.55; 0.11; 77; 0.11; 0.55]. In every single case, the extra control helps the perturbed system reach the desired steady states.

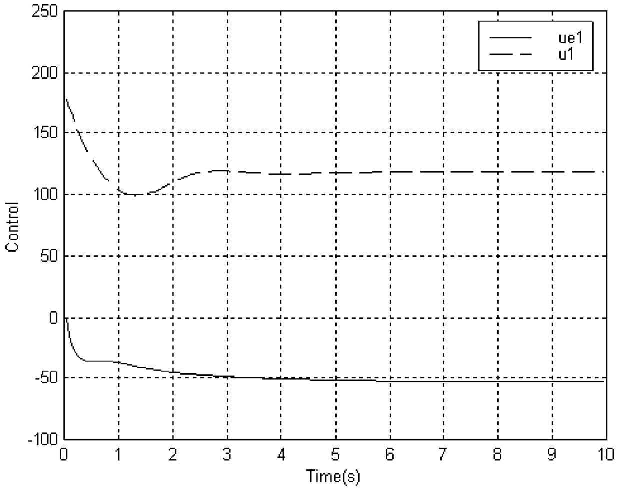

Fig. 17 Control from the throttle, u_1: 10% increase in initial values.

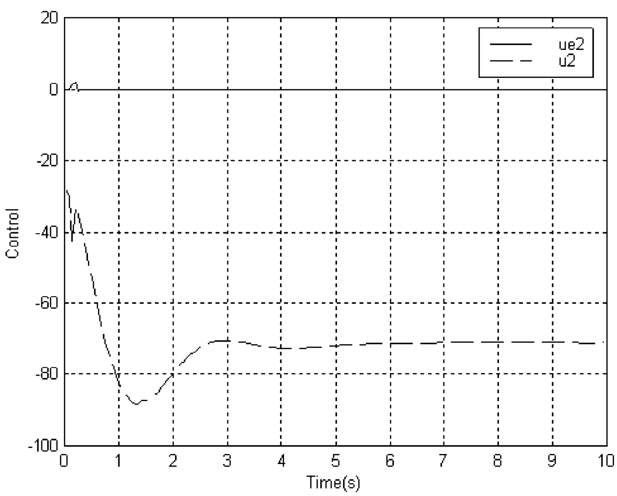

Fig. 18 Control from the collective servomechanism, u_2: 10% increase in initial values.

V. Conclusions

An NN solution to optimal control of nonlinear systems with applications to a helicopter was presented. The tracking histories showed that it is a viable solution. An extra control design through a Lyapunov analysis was also developed to counter the uncertainties in the basic model. Numerical results from the helicopter application showed that the design with the adaptive critic control plus the extra control is a sound technique for achieving practical stability.

Acknowledgments

This study was supported in part by a grant from the National Science Foundation, ECS-9976588 and a grant from EMARC, University of Missouri-Rolla. The authors are grateful to XiaHua Wang for help with some proofs.

References

[1]Bryson, A. E., and Ho, Y., *Applied Optimal Control*, Hemisphere, New York, 1975, pp. 128–211.

[2]Brockett, R. W., "Feedback Invariants for Nonlinear Systems," *Proceedings of the 7th IFAC Congress*, Vol. 2, 1978.

[3]Meyer, G., Su, R., and Hunt, L. R., "Application of Nonlinear Transformations to Automatic Flight Control," *Automatica*, Vol. 20, No. 1, 1984, pp. 103–107.

[4]Hunt, L. R., Su, R., and Meyer, G., "Global Transformations of Nonlinear Systems," *IEEE Transactions on Control*, Vol. AC-28, No. 1, 1983, pp. 24–30.

[5]Bugajski, D. J., Enns, D. F., and Elgersma, M. R., "A Dynamic Inversion Based Control Law with Application to the High Angle of Attack Research Vehicle," *Proceeding of the AIAA Guidance, Navigation, and Control Conference*, AIAA, Washington, DC, 1990, pp. 20–22.

[6]Snell, S. A., Enns, D. F., and Garrard, W. L., "Nonlinear Inversion Flight Control for a Supermaneuverable Aircraft," *Journal of Guidance, Control, and Dynamics*, Vol. 15, No. 4, 1992, pp. 976–984.

[7]Buffington, J. M., Sparks, A. G., and Banda, S. S., "Full Conventional Envelope Longitudinal Axis Flight Control with Thrust Vectoring," *Proceedings of the American Control Conference*, IEEE Publications, Piscataway, NJ, 1993, pp. 415–419.

[8]Chen, L., and Narendra, K. S., "On a New Approach to the Design of Tracking Controllers for Nonlinear Dynamical Systems," *Proceeding of American Control Conference*, IEEE Publications, Piscataway, NJ, 1999, pp. 3534–3538.

[9]Sontag, E. D., "Smooth Stabilization Implies Coprime Factorization," *IEEE Transactions on Automatic Control*, Vol. 34, No. 4, 1989, pp. 435–443.

[10]Sontag, E. D., and Wang, Y., "On Characterization of the Input-to-State Stability Property," *System and Control Letters*, Vol. 24, No. 5, 1995, pp. 351–359.

[11]Van Der Schaft, A. J., "On a State Space Approach to Nonlinear H_∞ Control," *System and Control Letters*, Vol. 16, No. 1, 1991, pp. 1–8.

[12]Van Der Schaft, A. J., "On the Hamilton–Jacobi Equation of Nonlinear H_∞ Control," *Proceedings of the 1st European Control Conference*, Edition Hèrmes Science Publications, Paris, 1991, pp. 649–654.

[13]Van Der Schaft, A. J., "L2-Gain Analysis of Nonlinear Systems and Nonlinear State Feedback H_∞ Control," *IEEE Transactions on Automatic Control*, Vol. 37, No. 6, 1992, pp. 770–784.

[14]Khalil, H. K., *Nonlinear Systems*, Prentice-Hall, Upper Saddle River, NJ, 1996, pp. 588–600.

[15]Balakrishnan, S. N., and Biega, V., "Adaptive Critic Based Neural Networks for Aircraft Optimal Control," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 4, 1996, pp. 893–898.

[16]Prokhorov, D., and Wunsch, D., "Adaptive Critic Designs," *IEEE Transactions on Neural Networks*, Vol. 8, No. 5, 1997, pp. 997–1007.

[17]Werbos, P. J., "An Overview of Neural Networks for Control," *IEEE Control Systems Magazine*, Vol. 11, No. 1, 1991, pp. 40, 41.

[18]Narendra, K. S., and Parthasarathy, K., "Identification and Control of Dynamical System Using Neural Networks," *IEEE Transactions on Neural Networks*, Vol. 1, No. 1, 1990, pp. 4–27.

[19]Polycarpou, M. M., and Ioannou, P. A., "Modelling, Identification and Stable Adaptive Control of Continuous-Time Nonlinear Dynamical Systems Using Neural Networks," *Proceedings of the American Control Conference*, IEEE Publications, Piscataway, NJ, 1992, pp. 36–40.

[20]Sanner, R. M., and Slotine, J. E., "Gaussian Network for Direct Adaptive Control," *IEEE Transactions on Neural Networks*, Vol. 3, No. 6, 1992, pp. 837–863.

[21]Lewis, F. L., Yesildirek, A., and Liu, K., "Multilayer Neural Net Robot Controller with Guaranteed Tracking Performance," *IEEE Transactions on Neural Networks*, Vol. 7, No. 2, 1996, pp. 388–399.

[22]Kim, B. S., and Calise, A. J., "Nonlinear Flight Control Using Neural Networks," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 1, 1997, pp. 26–33.

[23]Leitner, J., Calise, A. J., and Prasad, J. V. R., "Analysis of Adaptive Neural Networks for Helicopter Flight Control," *Journal of Guidance, Control, and Dynamics*, Vol. 20, No. 5, 1997, pp. 972–979.

[24]McFarland, M. B., Rysdyk, R. T., and Calise, A. J., "Robust Adaptive Control Using Single-Hidden-Layer Feedforward Neural Networks," *Proceeding of the American Control Conference*, IEEE Publications, Piscataway, NJ, 1999, pp. 4178–4182.

[25]Johnson, E. N., Calise, A. J., and El-Shirbiny, H. A., "Feedback Linearization with Neural Network Augmentation Applied to X-33 Attitude Control," AIAA Paper 2000-4157, Aug. 2000.

[26]Sutton, R. S., Barto, A. G., and Williams, R. J., "Reinforcement Learning in Direct Adaptive Optimal Control," *Proceedings of the American Control Conference*, IEEE Publications, Piscataway, NJ, 1991, pp. 2143–2146.

[27]Liu, X., and Balakrishnan, S. N., "Convergence Analysis of Adaptive Critic Based Optimal Control," *Proceeding of the American Control Conference*, IEEE Publications, Piscataway, NJ, 2000, pp. 1929–1933.

[28]Sira-Ramirez, H., Zribi, M., and Ahmad, S., "Dynamical Sliding Mode Control Approach for Vertical Flight Regulation in Helicopters, *IEE Proceedings on Control Theory and Applications*, Vol. 141, No. 1, 1994, pp. 19–24.

[29]Barron, A. R., "Neural Net Approximation," *Proceedings of the Seventh Yale Workshop on Adaptive and Learning System*, Durham Lab., Yale Univ., New Haven, CT, 1992, pp. 69–72.

[30]Lin, Z. L., and Knospe, C., "A Saturated High Gain Control for a Benchmark Experiment," *Proceedings of American Control Conference*, IEEE Publications, Piscataway, NJ, 2000, pp. 2644–2648.

[31]Mounfield, W. P., Jr., and Grujic, L. T., "High-Gain PI Control of an Aircraft Lateral Control System," *International Conference on Systems, Man and Cybernetics*, IEEE Publications, Piscataway, NJ, 1993, pp. 736–741.

[32]Rysdyk, R.T., and Calise, A. J.,"Nonlinear Adaptive Flight Control Using Neural Networks," *IEEE Control Systems Magazine*, Vol. 18, No. 6, 1998, pp. 14–25.