1-1-1995

Hamiltonian Based Adaptive Critics for Aircraft Control

S. N. Balakrishnan
Missouri University of Science and Technology, bala@mst.edu

Jie Shen

HAMILTONIAN BASED ADAPTIVE CRITICS FOR AIRCRAFT CONTROL

S. N. Balakrishnan and Jie Shen
Dept. of Mechanical and Aerospace Engineering
and Engineering Mechanics
University of Missouri-Rolla
Rolla, MO 65401-0249
(314) 341-4675

I. INTRODUCTION

Aircraft designs are becoming more complex in order to either operate more efficiently as in the commercial sector or to push the performance envelopes as in the military sector. An interesting outcome of this scenario is that 'control' has come to play an important role in helping realize these objectives. Increasingly, control is becoming an integral part of an aircraft design rather than an afterthought. More effective and efficient control of an aircraft is certain to lead to improved performance at a lower cost.

One other consequence of complexity in design is the need to consider innovative controller designs since existing designs are bound to be inadequate (for example, in high-angle of attack flights). We investigate the use of neural networks in this paper to formulate useful and efficient controllers. Controllers, in general, are designed to fulfill one of two functions. The first is called 'regulation whereby the controller drives the errors in a system states to zero. The second function of a controller is to track a reference signal (such as a desired pitch rate, etc.) within bounds. In either case, the designed controller should be able to operate successfully when there are realistic (expected) parameter variations within the system being controlled (robustness). In the evolution of control design, the classical control deals with robust controllers but they are essentially single-input, single-output (SISO) controllers. They are not quite optimal. Optimal controllers can handle multiple-input, multiple-output (MIMO) directly. However, robustness of an observer based optimal controller is not guaranteed. A lot of current research is devoted to build robust MIMO controllers based on some kind of optimization. It is hoped that in this area the field of neural networks can play a significant role. We investigate the use of neural networks to synthesize optimal controllers in this study. (The robustness studies currently under progress will be reported later.) We formulate a Hamiltonian based 'adaptive critic' (Figure 1) which provides optimal control for a wide range of initial conditions. Unlike other neural network solutions, this system of networks generates its own targets for their training. The 'adaptive critic' system is based on reinforcement learning and consists of two neural networks (if the model is known). One element, called 'the critic,' provides an assessment of the second element called 'the action' (control) network so that its outputs are 'better' in the future [3,4]. We use this concept to solve the Hamiltonian based control equations associated with optimal control. In the remainder of the paper, we present the general optimal control problem and the associated equations in Section II. A scalar problem in order to demonstrate the procedure in adaptive critic based solutions is presented in Section III. A typical aircraft control problem is also presented in Section III. The conclusions are summarized in Section IV.

II. PROBLEM FORMULATION AND SOLUTION DEVELOPMENT

Cost Function

ISBN# 0-7803-2636-9

Through the neural network methodology presented in this study, we will be able to solve a class of optimal control problems. The cost function in such cases is given by, J, where

$$J = \phi[x(N)] + \sum_{i=0}^{N-1} L_i\,[x(i),\, u(i)]. \qquad (1)$$

In Eqn. (1), $L_i()$ can be a linear or nonlinear function of the states and/or control and $\phi()$ can be a linear or nonlinear function of terminal states. I indicates the stage. The underlying system model is given by

$$x(i+1) = f_i(x(i),\, u(i)) \qquad (2)$$

where $f_i()$ can be either linear or nonlinear. The optimal control problem can be formulated in terms of Hamiltonian [1] where the Hamiltonian, H_i, is given by

$$H_i = L_i(x(i),\, u(i)) + \lambda^T(i+1)\, f_i(x(i)),\, u(i) \qquad (3)$$

The propagation equations for the Lagrange Multiplier, i = 0,1, ...N-1, are given by

$$\lambda(i) = (\partial f_i/\partial x(i))^T\, \lambda(i+1) + (\partial L_i/\partial x(i))^T \qquad (4)$$

with boundary condition on λ as

$$\lambda(N) = (\partial\phi/\partial x(N))^T \qquad (5)$$

The optimality condition is

$$\partial H_i/\partial u(i) = 0,\ i = 0,\ l,\ \ldots,\ n-1 \qquad (6)$$

Note that for a steady state regulator problem $\phi()$ is zero and N→ large.

Adaptive Critic

The goal of the neural networks is to find the control which minimizes the cost in Eqn. (1) by solving Eqn. (2) and (4) with the use of Eqn. (6) and boundary conditions given by Eq. (5) and the known initial states.

In order to accomplish this task, we use two networks. One network called 'action' models the control, u(i), for which the inputs are x(i). In order to train the control network, first, x(i), is randomized and the action network outputs u(i). The system model in Eqn. (2) is then used to find x(i+1). The derivatives $\partial f_i/\partial x(i)$ and $\partial L_i/\partial x(i)$ can all be calculated since x(i) and u(i) are known. Now, a randomized critic network is considered and $\lambda(i)$ and $\lambda(i+1)$ are calculated corresponding to x(i) and x(i+1). With $\lambda(i+1)$, the target $\lambda(i)$, denoted by $\lambda*(i)$, can be calculated by using Eqn. (4). The difference between $\lambda*(i)$ and $\lambda(i)$ 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. This is done by finding u(i) for random x(i) and correcting them through the use of the model equation in Eqn. (2) to find x(i+1), and use x(i+1) to find $\lambda(i+1)$ from the critic network corresponding to x(i+1). By using $\lambda(i+1)$ in Eqn. (6), we can solve for the target $u*(i)$ and use it to correct the action network.

This two-step procedure continues till a predetermined level of convergence is reached.

III. APPLICATIONS

In this section of the study, two specific examples will be dealt with. The first of these is an infinite horizon one dimensional linear problem. After this motivating example, a four-dimensional aircraft control problem is presented.

A. Infinite Time 1-D Linear Application

The first application deals with a problem of the form

$$x(i+1)=2x(i)+3u(i) \qquad (7)$$

and a cost function of the form

$$J=\sum_{i=0}^{\infty} [2x^2(i)+2u^2(i) + 1.1x(i)\,u(i)] \qquad (8)$$

As a first step in the solution, a stabilizing controller is defined. In the case of this problem, the initial control is defined as

$$u(i) = -0.4x(i)\,. \tag{9}$$

Alternately, the control can be initiated by a network with random weights. Next, a neural network is designed and the initial weights of this network are randomized. For this problem, the network has three layers and each of the hidden layers possesses three neurons. This network functions as the critic.

It can be observed that for the infinite horizon problem the cost associated with state x(i) at time t should be equal to the cost associated with state x(i) at time t+1; therefore, a single critic can be used to calculate both λ(x(i)) and λ(x(i+1)). The Hamiltonian, H_i, in this case is

$$H_i(x(i),\, u(i)) = x^2(i) + u^2(i) + 2x(i)\,u(i) + \lambda(i+1)\,(x(i) + 2\,u(i)) \tag{10}$$

Note that we can obtain the derivatives of the Hamiltonian from Eqn. 6. This, in combination with the critic outputs and the system model derivatives, allows the use of Eqn. 4 to determine the target value for the critic λ^*(x(i)). This target value is calculated for random values of x(i) until the critic network converges.

After the critic converges, a new neural network is initialized to act as the action network. For this problem a neural network with two hidden layers and three neurons per layer is chosen. The action network is then trained using a gradient descent algorithm with outputs from the converged critic network which are used in solving Eqn. 6 for control.

After the action network converges, the critic is again trained using the outputs of the new action network. (Note that the weights of the critic are not randomized. Instead, the weights from the previous critic are used as the initial weights.) This process is repeated until both networks converge. At this point, the outputs of the action network produce optimal control.

The evolution of the control law (or the action network) is presented in Figure 2. The square solid line represents the assumed control used in the design of the first critic. The corresponding critic is presented by a square line in Figure 3. Other curves in Figure 3 represent the evolution of the critic. In three iterations of the action and critic networks, both networks have shown close convergence as can be observed from Figures 2 and 3. At this point, the action network is expected to output control which is optimal (even though we have presented a few more iterations). In order to check the optimality of the output, the optimal control law obtained through a Ricatti solution formulation [1] is also shown in Figure 4. It is observed that at almost all the points considered, the neural network based control is nearly identical with the Ricatti solution.

A similar comprehensive controller was introduced by Balakrishnan and Biega [4]. In that paper, they used a dynamic programming based adaptive critic to produce optimal control. The results of the formulation in [4] for the same scalar problem are presented in Fig. [4]. Note that it takes ten iterations for the networks to output optimal control for the dynamic programming based critic while the Hamiltonian based critic is close to the optimal control in only three iterations.

Figure 5 shows a comparison of the system state being controlled by both the optimal control (Ricatti solution) and the control determined by this adaptive critic based method for x(0) = -20. Note that this initial condition was chosen arbitrarily. The neural network has determined the near optimal control law for each point within its training range.

B. Aircraft Control Application

We consider the synthesis of an optimal longitudinal autopilot in this section. The performance index in this application is an infinite-time quadratic cost function. The minimizing control is expected to drive the deviations of the longitudinal dynamics in pitch angle, θ, pitch rate, q, forward velocity, u', and angle of attack, α, to zero.

The orientation of an aircraft involving longitudinal dynamics is shown in Figure 6. The linearized equations of motion of an aircraft in a vertical plane are given by

$$\dot{x} = Ax + Bu \quad (11)$$

where the elements of the state space **x** are

$$x \equiv [u', \alpha, \theta, q]^T. \quad (12)$$

Elements a_{ij}, i=1,2,3,4, j= 1,2,3,4 of the 4x4 matrix, A, represent the dynamic stability derivatives and are given by

$$\begin{aligned} &a_{11} = -0.0148,\ a_{12} = -13.88,\ a_{13} = -32.2,\ a_{14} = 0 \\ &a_{21} = -0.00019,\ a_{22} = -0.84,\ a_{23} = 0,\ a_{24} = 1, \\ &a_{31} = 0,\ a_{32} = 0,\ a_{33} = 0,\ a_{34} = 1 \\ &a_{41} = 0.00005,\ a_{42} = -4.8,\ a_{43} = 0,\ a_{44} = -0.5. \end{aligned} \quad (13)$$

Elements b_{ij}, i= 1,2,3,4, j=1 of the 4x1 matrix **B** represent the control derivatives and are given by

$$b_{11} = -1.1 \quad b_{21} = -0.11 \quad b_{31} = 0 \text{ and } b_{41} = -8.74. \quad (14)$$

The control variable **u** represents elevator deflection.

The performance index, **J**, is formulated so as to keep the pitch angle, pitch rate, normal acceleration and elevator deflection low and penalize if they exceed the prespecified maximum values. That is,

$$J = \int_0^{\infty} \left[(\theta/\theta_{max})^2 + (q/q_{max})^2 + (n_z/n_{z_{max}})^2 + (u/u_{max})^2 \right] dt \quad (15)$$

where

$$\theta_{max} = 0.26 \text{ rad}, \quad g_{max} = 0.31 \text{ rad/sec},$$
$$n_{z_{max}} = 6 \text{ g's and } u_{max} = 0.1 \text{ rad}$$

n_z represents normal acceleration and, g, is the gravitational acceleration which is set at 32.2 ft/sec^2. Note that n_z can be obtained in terms of other states as

$$n_z = (Vo/g)(q - \dot{\alpha}) \quad (16)$$

where Vo is the steady state aircraft velocity.

Note that this performance index, with the use of Eqns. 15 and 16 has the form

$$J = \int_0^{\infty} (x^T Q x + u^T R u + 2x^T P u)\, dt \quad (17)$$

where Q, R and P are appropriate weighting matrices in terms of θ_{max}, q_{max}, $n_{z_{max}}$, and u_{max}. These are

$$R = 91.32, \ P = (0, 2.2*E{-}4, 0.97, 0)^T$$

and

$$Q = \begin{bmatrix} 10.37 & 0 & 0 & 0 \\ 0 & 3.7*E{-}7 & 1.65*E{-}3 & 0 \\ 0 & 1.65*E{-}7 & 7.25 & 0 \\ 0 & 0 & 0 & 14.84 \end{bmatrix}.$$

Note that P is present because of cross terms in x and u which occur after n_z is rewritten in terms of state equations. Solutions to this optimization problem is obtained using the adaptive critic approach described in the last section.

The numerical results from these experiments are presented in Figures 7-12. Histories of $u'(t)$, $\alpha(t)$, $\theta(t)$, and $q(t)$ with time are presented in Figures 8-11, respectively. In order to demonstrate the versatility of the adaptive critic approach, we have presented plots of the neural network-based states and optimal state histories for <u>arbitrary</u> initial conditions in Figures 7-10. In <u>each</u> one of the cases, we can observe that the optimal trajectories (from exact Ricatti solutions) and the neural network based solutions are virtually identical. It should be observed that all these control outputs are generated from <u>one</u> converged neural network. In other words, the action network can be used as one repository of gains for various operating conditions or errors. It is a feedback controller since the inputs are the current states and the outputs are the control values. Note that <u>no</u> external training is necessary to achieve this. The optimal control history and the neural network based control history are presented in Figure 12.

IV. CONCLUSIONS

A new Hamiltonian based adaptive critic architecture to solve optimal control problems has been presented. A scalar problem has been solved to illustrate the steps in the design process of the dual network structure. It has been shown that these networks can produce near optimal control policies for infinite horizon problems such as an aircraft control problem. This architecture requires no external training data and yields optimal control through the entire range of operation and can be used in closed loop. Since the controller network contains an envelope of gains, it can act as an autopilot. The added advantage of this approach is that the critic network can provide fault tolerance. Future work on this topic will investigate the robustness of such network controllers and the use of this method for finite-horizon class of problems.

ACKNOWLEDGEMENT

This study was funded by NSF Grant ECS-9313946 and by the Missouri Department of Economic Development Center for Advanced Technology Program. The authors thank Dr. Paul Werbos for his technical interactions during this study.

BIBLIOGRAPHY

[1]. Bryson, A. E. and Ho., Y., *Applied Optimal Control*, Hemisphere Publishing Co., 1975, pp. 128-211.

[2]. White, D. A. and Sofge, D., *Handbook of Intelligent Control*, Van Nostrand Reinhold, 1992, Ch. 3,5,8,12,13.

[3]. Werbos, P., "Neurocontrol and Supervised Learning; An Overview and Evaluation," *Handbook of Intelligent Control*, Van Nostrand Reinhold, 1992.

[4]. Balakrishnan, S. N. and Biega, V., "Adaptive Critic Based Neural Network for Aircraft Control," submitted for review to *AIAA Journal of Guidance, Control, and Dynamics*.

Figure 1. Adaptive Critic for Control

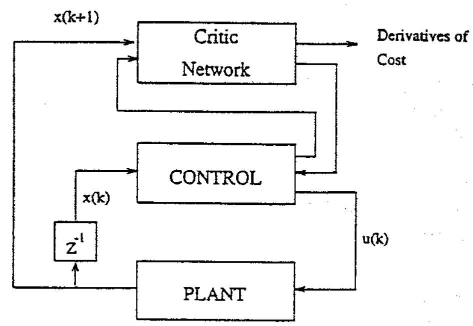

Figure 2. Evolution of Controller Network

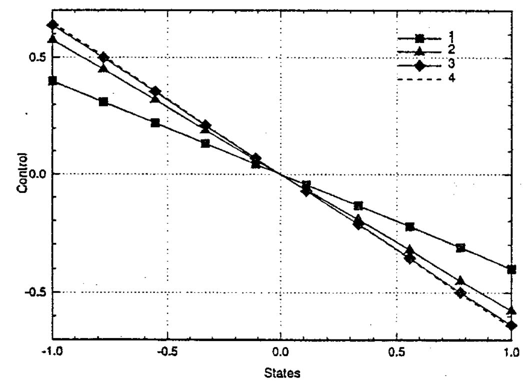

Figure 3. Evolution of Critic Network

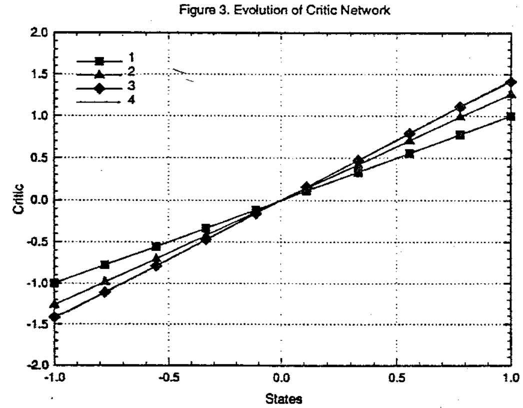

Figure 4. Controller Network (Dynamic Programming)

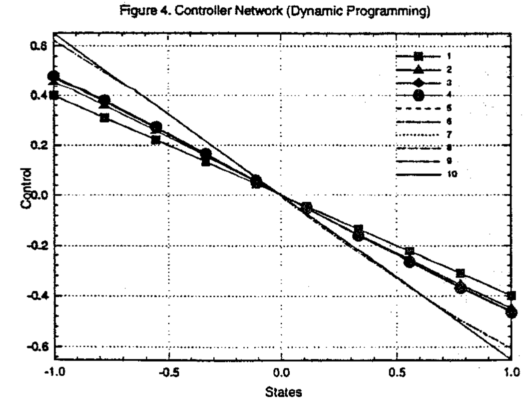

Figure 5. Comparison of Trajectories

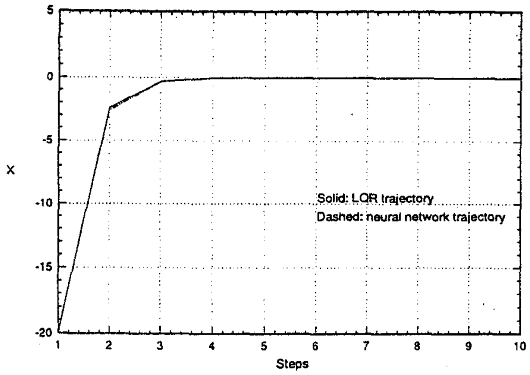

Figure 6. Aircraft Control Scenario

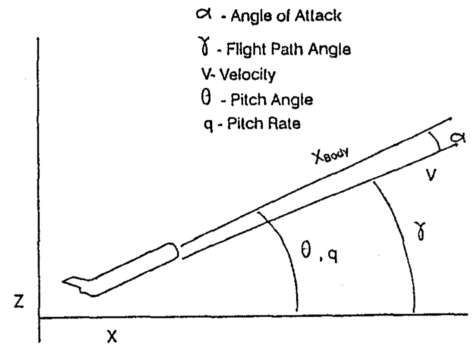

Figure 7. History of u'(i) for various initial conditions

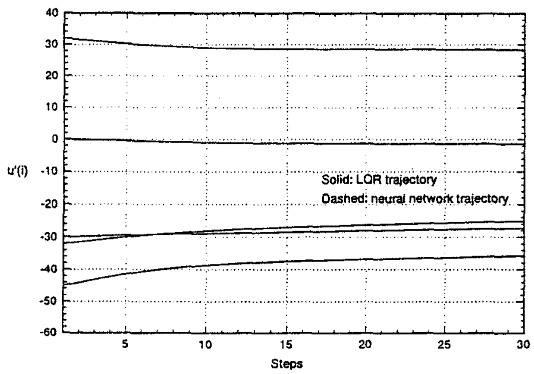

Figure 8. History of α(i) for various initial conditions

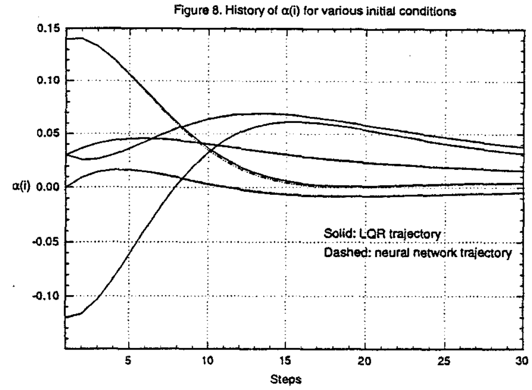

Figure 9. History of q(i) for various initial conditions

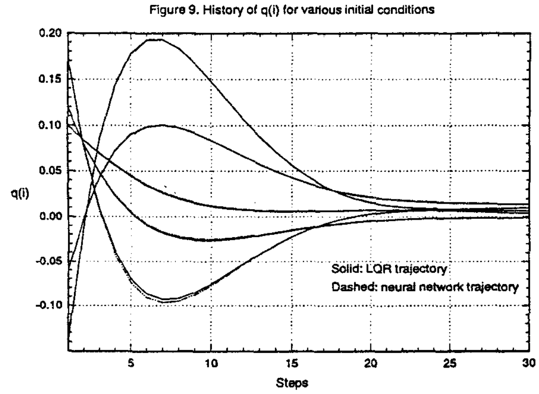

Figure 10. History of θ(i) for various initial conditions

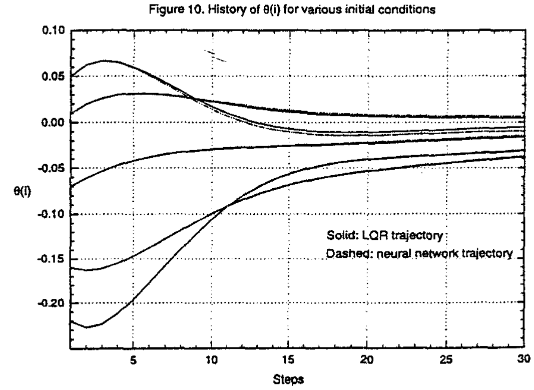

Figure 11. History of u(i) for various initial conditions

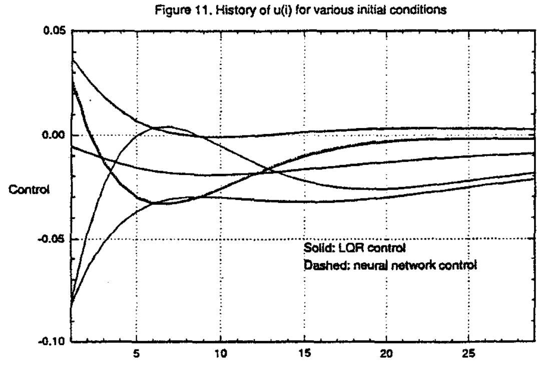

Figure 12. Error history between neural network control and LQR control for various initial conditions