An Adaptive Critic Approach to Reference Model Adaptation

K. KrishnaKumar[†], G. Limes[*], K. Gundy-Burlet[†], D. Bryant[&]

Abstract

Neural networks have been successfully used for implementing control architectures for different applications. In this work, we examine a neural network augmented adaptive critic as a Level 2 intelligent controller for a C-17 aircraft. This intelligent control architecture utilizes an adaptive critic to tune the parameters of a reference model, which is then used to define the angular rate command for a Level 1 intelligent controller. The present architecture is implemented on a high-fidelity non-linear model of a C-17 aircraft. The goal of this research is to improve the performance of the C-17 under degraded conditions such as control failures and battle damage. Pilot ratings using a motion based simulation facility are included in this paper. The benefits of using an adaptive critic are documented using time response comparisons for severe damage situations.

1 Introduction

In the last 30 years, at least 10 aircraft have experienced major flight control system failures claiming more than 1100 lives[1, 2]. The Intelligent Flight Control (IFC) research program at NASA Ames began in 1992 to address adaptive aircraft control. The major feature of IFC technology is its ability to adapt to unforeseen events through the use of self-learning neural flight control architecture. These events can include sudden loss of control surfaces, engine thrust, and other causes that may result in the departure of the aircraft from safe flight conditions.

To provide a real-time system capable of compensating for a broad spectrum of failures, NASA researchers have investigated a neural flight control architecture[3, 4], developed by Rysdyk and Calise[5], for both flight and propulsion control. The concept was to develop a system capable of utilizing all remaining sources of control power after damage or failures. The Integrated Neural Flight and Propulsion Control System (INFPCS) uses a daisy-chain control allocation technique to ensure that conventional flight control surfaces will be utilized under normal operating conditions. Under damage or failure conditions, the system may allocate flight control surfaces, and incorporate propulsion control, when additional control power is necessary for achieving desired flight control performance.

The NASA Ames Intelligent Flight Controller (IFC) uses a neural flight control architecture that is based upon the augmented model inversion controller. This direct adaptive tracking dynamic inverse controller integrates feedback linearization theory with both pre-trained and on-line learning neural networks. Pre-trained neural networks are used to provide estimates of aerodynamic stability and control characteristics required for model inversion. On-line learning neural networks are used to generate command augmentation signals to compensate for errors in the estimates and from the model inversion. The on-line learning neural networks also provide additional potential for adapting to changes in aircraft dynamics due to damage or failure. Reference models are used to filter command inputs in order to specify desired handling qualities. A Lyapunov stability proof guarantees boundedness of the tracking error and network weights[5]. Successful piloted simulation studies were also performed at NASA Ames Research Center on a commercial transport aircraft simulator[3]. Subjects included both NASA test pilots and commercial airline crews.

The research reported in this paper is an extension of the above work to include reference model adaptation using adaptive critic technologies[6,7,14] and validation of this approach using a non-linear C-17 simulation test bed. Reference model adaptation is considered a higher level of intelligence in the hierarchy of intelligent control[8,12,13]. It is necessary in situations where the system is degraded to a level where control authority is insufficient to achieve the required reference model characteristics. The idea then is to degrade the reference model to match the capabilities of the airplane. This is achieved using the adaptive critic technology. Adaptive critics have been successfully used in a reference model application for an engine control problem[14]. Other applications of adaptive critics include aircraft control[9,10] and spacecraft control[11].

This paper presents overviews of adaptive critic technology and intelligent flight control architecture based on the idea of levels of intelligent flight control[8,12,13]. Next, we elaborate the adaptive critic implementation for reference model adaptation. Finally, the results of the application to aircraft control problem are discussed.

2 Levels of Intelligent Control

Over the past decade, several innovative control architectures utilizing the intelligent control tools have

[†] Research Scientist, NASA Ames Research Center, Moffett Field, CA

[*] Computer Engineer, QSS Inc. Ames Research Center, Moffett Field, CA

[&] Pilot, QSS Inc. Ames Research Center, Moffett Field, CA been proposed. KrishnaKumar[8,12,] has proposed a classification scheme based on the ability of the intelligent flight control architecture for self-improvement (see Table 1). The classification scheme divides the control architectures among levels of intelligent control. For instance, most of the proposed architectures can be divided among level 0, level 1, level 2, and level 3 intelligent control schemes. Based on this classification scheme, several seemingly differing control architectures can be looked at as achieving similar goals.

Table 1. The Levels of Intelligent Control

L	Self improvement of	Description
0	Tracking Error (TE)	Robust Feedback Control: Error tends to zero.
1	TE + Control Parameters (CP)	Adaptive Control: Robust feedback control with adaptive control parameters (error tends to zero for non-nominal operations; feedback control is self improving).
2	TE + CP + Performance Measure (PM)	Optimal Control: Robust, adaptive feedback control that minimizes or maximizes a utility function over time.
3	TE+CP+PM + Planning Function	Planning Control: Level 2 + the ability to plan ahead of time for uncertain situations, simulate, and model uncertainties.

3 Adaptive Critics

Adaptive critic designs have been defined as designs that attempt to approximate dynamic programming based on the principle of optimality. Adaptive critic designs consist of two entities, an action network that produces optimal actions and an adaptive critic that estimates the performance of the action network. The adaptive critic is an optimal or near optimal estimator of the cost-to-go function that is trained (adapted) using recursive equations derived from dynamic programming. The critic is termed adaptive as it adapts itself to output the optimal cost-to-go function from a given system state. The action network is adapted simultaneously based on the information provided by the critic. The action network consists of any piece of the overall control architecture that has an effect on the final performance of the closed-loop system. In typical applications, the action network consists of the controller that is optimized using the critic. The inputs required for designing an adaptive critic design are

- The cost function or the performance measure.
- A parameterized representation of the critic.
- A parameterized representation of the action network.
- A method for adapting the parameters of the critic.

3.1 Choice of the cost function

The choice of the cost function comes from the problem at hand. The cost could be distributed over the entire length of time or be defined at the end of the process. Typical examples of the two types are minimizing the fuel spent for a certain flight mission or intercepting a projectile where the utility depends only on the final error. Typically, the cost function can be given as,

$$(1) \qquad J=\sum_{i=0}^{T}\gamma^{i}U[x(i),u(i)]$$

where $U[x(i),u(i)]$ is the utility function or a penalty function that is a function of the state of the system, *x(i)*, and the control (action), *u(i)*, given to the system. 'γ' is a discount factor that discounts the future performance.

The dynamic programming principle states that we can formulate an optimal control problem where we can get an optimal solution by minimizing the cost-to-go function, *J(t),* which is defined as,

$$(2) \qquad J(t)=\sum_{i=1}^{T-t}\gamma^{i}U[x(t+i),u(t+i)]$$

So the critic is designed to approximate the optimal form of this cost-to-go function or its derivatives with respect to the state of the system depending on the particular adaptive critic design.

3.2 Parameterized representation for the critic and the action network

Parameterization of the critic and the action network is achieved by the use of neural networks. Having learnt to model a system, neural networks can be used to provide sensitivities of the system outputs with respect to the system inputs. This proves to be useful information especially for the training of our intelligent control architecture. Reference 6 provides detailed insight into the area of neural networks and their use in control.

3.3 Training the critic

Several methods have been proposed for training adaptive critics that are based on the dynamic programming equation. These methods vary based on the level of complexity and the degree of accuracy sought for training the cost-to-go function. Some of these methods are the heuristic dynamic programming (HDP) approach, the dual heuristic programming (DHP) approach, and the global dual heuristic dynamic (GDHP) programming approach. The DHP approach was used in this study and is outlined in section 5.1 References 6-7 provide a more detailed discussion on the subject.

4 NASA Ames Intelligent Flight Control Architecture

Adaptive critic methods have been successfully applied to IFC architectures[9-12,14]. Figure 1 presents the implementation of a Level 2 Intelligent Controller on a C-17 test bed used in this study. The levels of intelligent control outlined earlier are labeled in the figure. It should be noted that Level 0 is non-adaptive whereas Level 1 is adaptive. Level 1 is non-optimal whereas Level 2 is optimal. Details of the figure are presented below.

Reference Models: The pilot commands roll and pitch rates and aerodynamic lateral accelerations through stick and rudder pedal inputs. These commands are then transformed into body-axis rate commands, which also include turn coordination, level turn compensation, and yaw-dampening terms. First-order reference models are used to filter these commands in order to shape desired handing qualities.

P + I Error Controller (Level 0): Errors in roll rate, pitch rate, and yaw rate responses can be caused by inaccuracies in aerodynamic estimates and model inversion. Unidentified damage or failures can also introduce additional errors. In order to achieve a rate-command-attitude-hold (RCAH) system, a proportional-integral (PI) error controller is used to correct for errors detected from roll rate, pitch rate, and yaw rate (p, q, r) feedback.

Learning Neural Network (Level 1): The on-line learning neural networks work in conjunction with the error controller. By recognizing patterns in the behavior of the error, the neural networks can learn to remove biases through control augmentation commands. These commands prevent the integrators from having to windup to remove error biases. By allowing integrators to operate at nominal levels, the neural networks enable the controller to provide consistent handling qualities. The learning neural networks not only helps control the nominal system, but also provides an additional potential for adapting to changes in aircraft dynamics due to control surface failures or airframe damage.

Dynamic Inversion/Aero Generation: The dynamic inversion element converts the summed response commands into virtual control surface commands. Dynamic inversion is based upon feedback linearization theory. No gain-scheduling is required, since gains are functions of aerodynamic stability and control derivative estimates and sensor feedback. Several methods are available to accomplish approximate model definition: simple linear model methods, nonlinear tables or using pre-trained neural networks (non-changing) to provide estimates of aerodynamic stability and control characteristics. The model is then inverted to solve for the necessary control surface commands. In our work, a Levenberg-Marquardt (LM) multi-layer perceptron[17] is used to provide dynamic estimates for model inversion. The LM network is pre-trained with stability and control derivative data generated by a Rapid Aircraft Modeler, and vortex-lattice code[15].

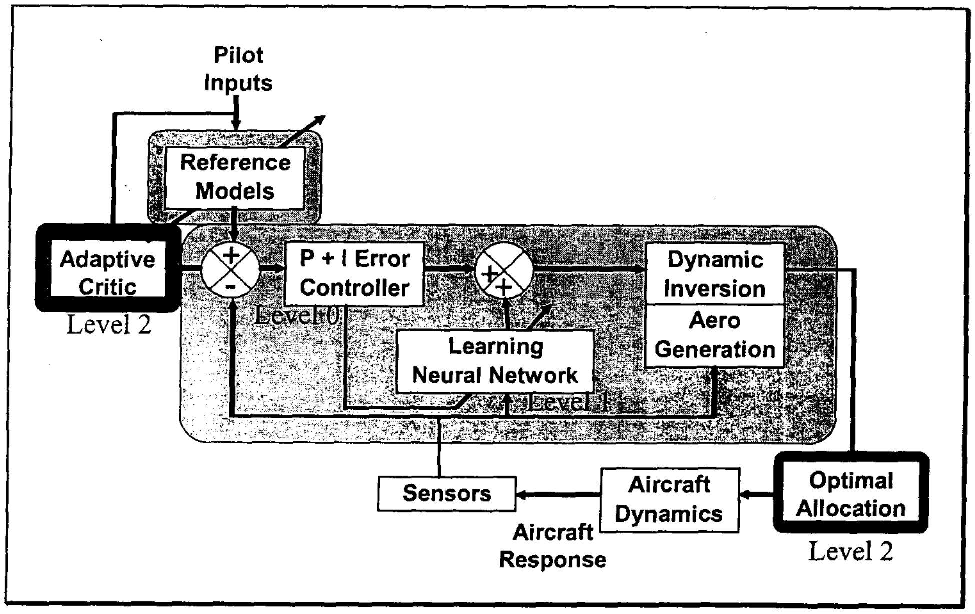

Figure 1. Intelligent flight control architecture

Optimal Allocation (Level 2): This system uses a linear programming technique to optimally allocate required acceleration to available control surfaces based on perceived limits. A best control allocation hierarchy is defined using a set of weightings on individual control usage. The weights depend on the allowed structural limits and user's preferences. These pre-programmed weights will drive the optimal control allocation based on a linear programming formulation. Unconventional flight control surface allocations are only utilized when the primary flight control surface commands exceed the known limits of deflection. For example, yaw rate control is normally provided through rudder deflection. If this command should saturate, then the remaining portion of the command is applied via a blended solution that could result in the deflection ailerons and/or spoilers. This decision is optimally chosen by the control allocation technology based on linear programming.

Adaptive Critic (Level 2): In the event of a severe degradation in performance of an aircraft, pilot handling qualities as dictated by the reference model cannot be maintained. It will be desirable to "optimally" modify the dynamics of the reference model to suit the situation in hand. Towards these goals an adaptive critic is utilized to optimize the shape of the reference model dynamics in the event of a failure or damage.

5 Adaptive Critic Application

This paper highlights the adaptive critic application to reference model adaptation. A static reference model is sufficient when the system is functioning normally. The model needs to change when desired performance is not achievable with the available control authority. If this problem is not rectified, two issues arise: (1) wrong signal for NN training for Level 1; (2) error wind-up. The adaptive critic application to reference model adaptation addresses these issues. Another issue, although not considered in this paper, is the use of engines for rotational control. Engines are not fast enough to provide the same handling qualities. In cases where propulsion control is used, adaptive critics can be used to optimally adjust the reference model frequencies.

The critic is implemented as a Dual-heuristic Dynamic Programming (DHP) critic. The DHP scheme is similar in idea to the HDP scheme, however, in DHP, the critic outputs the derivative of the performance with respect to control directly, which is the signal necessary for adapting the controller. Though DHP is more complex both theoretically and in implementation, it is generally considered to produce better results.

5.1 Use of the Reference Model as the Action Network

The overall implementation of the adaptive critic architecture for the aircraft control problem is shown in Figure 2. In many of the methods using adaptive critics, the trained critic is finally used to update the controller. In this implementation, the reference model is looked at as the action network. In other words, the reference model is the input into the closed-loop system consisting of the Level 1 intelligent controller and thus is the right choice as the action network. In the face of any failures or damage, it is sometimes impossible for the controller to achieve the system outputs as demanded by the original reference model. The Level 2 intelligent controller using the adaptive critic neural network, therefore attempts to adapt the parameters of the reference model to provide the system realizable performance.

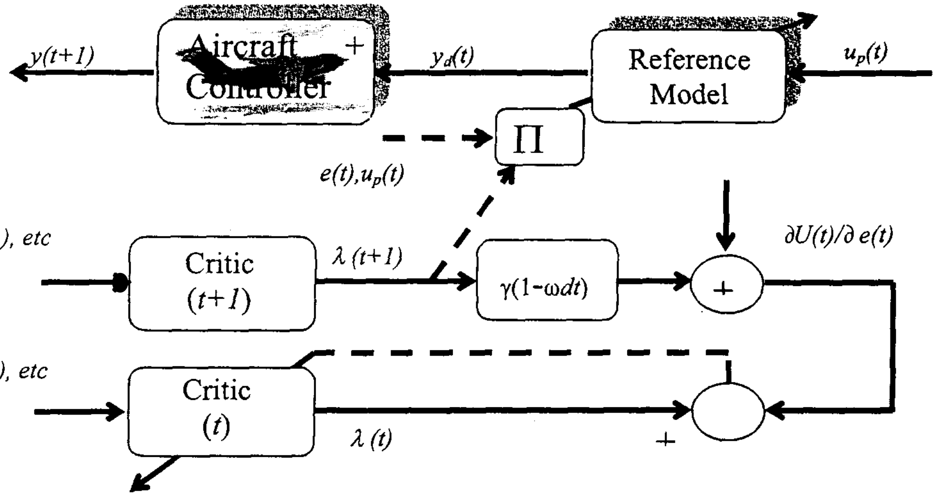

Figure 2. Reference model tuning using the adaptive critic

5.1.1 Adaptive Critic Neural Network

Adaptation is achieved using a single hidden-layer neural network with the following inputs:

- error in the corresponding axes at time *t*
- error rate in the corresponding axes at time *t*
- control needed beyond the limits at time *t*
- pilot inputs at time *t*

The neural network output is defined to be the derivative of the performance index with respect to the individual error in the axes (roll rate error, pitch rate error, and yaw rate error). The pertinent equations are given below:

$$J(t)=\sum_{t=T}^{\infty}U(t) \quad \Rightarrow \quad \lambda_e(t)=\left(\frac{\partial J}{\partial e}\right)_t$$

(3) $$U(t)=f(e^2)$$

$$f(e^2)=\frac{1}{1+\exp(-m(e^2-c))}$$

where

e = error in each of the three axes (p, q, r)
m, c = constants chosen by the user

A sigmoid function is chosen as a penalty function since it penalizes the performance measure only if the error rises beyond a certain value, *c*. It can be seen from equation 3, for a value of c = 0.01, the performance measure starts getting penalized when the error approaches 0.1 rad/sec. So, for values less than 0.1 rad/sec the penalty is close to zero and for values beyond 0.1 rad/sec the penalty is close to one. At the same time since the penalty function goes from zero to one, it automatically provides itself as a normalized function. The constant *m* defines the slope of the Sigmoidal curve.

5.1.2 DHP Critic Training Equations

We use a first order reference model for our problem. Given

(4) $$\dot{y}_d=-\omega_d y_d+k_d u_p,$$

subtracting $\dot{y}$ and $\omega_d y$ from both sides of the equation and rearranging we get

(Note: $y = actual\ p,q,r$).

(5) $$\begin{aligned}\dot{y}-\dot{y}_d&=-\omega_d(y-y_d)-k_d\omega_d u_p+(\dot{y}+\omega_d y)\\ \dot{e}&=-\omega_d e-k_d\omega_d u_p+(\dot{y}+\omega_d y)\\ \dot{e}&=-\omega_d e-k_d\omega_d u_p+d\hat{\delta}\end{aligned}$$

In the above equation, $d\hat{\delta}=(\dot{y}+\omega_d y)$ represents the additional control needed and could be seen in this context as an external disturbance that causes error to behave non-optimally.

In discrete form (first order forward differencing),

$$e_{t+1}-e_t=(-\omega_d e_t-k_d\omega_d u_{p,t}+d\hat{\delta})dt$$

$$e_{t+1}=(1-\omega_d dt)e_t-(k_d\omega_d dt)u_{p,t}+\dot{y}dt+\omega_d y dt$$

Now, from Dynamic Programming, we know that,

(6) $$\min_{\omega_d,k_d} J_t=\min_{\omega_d,k_d}(\gamma\ J_{t+1}+U_t),$$

where γ is a discount factor (usually > 0.9)

Now differentiating with respect to e_t and noting that minimization is an approximation and that J is only an estimate, we have

(7) $$\left(\frac{\partial\hat{J}_t}{\partial e_t}\right)_{desired}=\left[\hat{\lambda}_t\right]_{desired}=\gamma\ \frac{\partial\hat{J}_{t+1}}{\partial e_t}+\frac{\partial U_t}{\partial e_t}$$

with

(8) $$\frac{\partial U_t}{\partial e_t}=\frac{2*m*e*\exp(-m(e^2-c))}{(1+\exp(-m(e^2-c)))^2}$$

and

(9) $$\frac{\partial\hat{J}_{t+1}}{\partial e_t}=\frac{\partial\hat{J}_{t+1}}{\partial e_{t+1}}\frac{\partial e_{t+1}}{\partial e_t}=\left[\hat{\lambda}_{t+1}\right](1-\omega_d dt)$$

Implying

(10) $$\left(\frac{\partial\hat{J}_t}{\partial e_t}\right)_{desired}=\left[\hat{\lambda}_t\right]_{desired}$$

$$=\gamma\ \left[\hat{\lambda}_{t+1}\right](1-\omega_d dt)+\frac{\partial U_t}{\partial e_t}$$

The above quantity is the training signal for the critic at time “t”

5.1.3 Reference Model Adaptation Equations

Now we derive the computation of the performance sensitivities, ($\frac{\partial J}{\partial\omega_{d1}}$ and $\frac{\partial J}{\partial k_d}$), for adapting the reference model parameters, ω_{d1}, k_d. In the rest of the presentation, the notation '^' has been dropped for convenience.

$$\frac{\partial J_{t+1}}{\partial \omega_d} = \left[\frac{\partial J_{t+1}}{\partial e_{t+1}}\right]\left(\frac{\partial e_{t+1}}{\partial \omega_d} + \frac{\partial e_{t+1}}{\partial e_t}\frac{\partial e_t}{\partial \omega_d} + \frac{\partial e_{t+1}}{\partial e_t}\frac{\partial e_t}{\partial e_{t-1}}\frac{\partial e_{t-1}}{\partial \omega_d} + \ldots\right)$$

$$= -\lambda_{t+1}\left(f_t dt + (1-\omega_d dt) f_{t-1} dt + (1-\omega_d dt)^2 f_{t-2} dt + \ldots\right)$$

$$= -\lambda_{t+1}\left(\sum_{j=0}^{j=N} (1-\omega_d dt)^j f_{t-j} dt\right)$$

where

$$f_t = (e_t - y_t + k_d u_{p,t})$$

If we let $S_1 = f_0 dt$, one can write a dynamic equation for the quantities under summation as follows,

(11) $$S_{t+1} = (1-\omega_d dt)S_t + f_t dt, \quad \infty \le t \le 1$$

hence,

(12) $$\frac{\partial J_{t+1}}{\partial \omega_d} = -\lambda_{t+1} S_{t+1}$$

Similarly,

(13) $$\frac{\partial J_{t+1}}{\partial k_d} = -\lambda_{t+1} R_{t+1}$$

where,

(14) $$R_1 = \omega_d u_{p,0} dt$$

(15) $$R_{t+1} = (1-\omega_d)R_t + \omega_d u_{p,t} dt$$

A smoothing algorithm can be used to desensitize the adaptation to sensor noise and other unwanted variations in the error estimate. This is achieved using a smoothing algorithm given below:

(16) $$\frac{\partial J}{\partial \omega_d} = -\frac{1}{n+1}\sum_{t=t_{now}}^{t_{now}-n*dt} \lambda_t e_{t-dt} dt$$

(17) $$\frac{\partial J}{\partial k_d} = -\frac{1}{n+1}\sum_{t=t_{now}}^{t_{now}-n*dt} \lambda_t u_{p,t-dt} dt$$

Now the parameters of the reference model can be adapted using the following gradient descent equations

(18) $$\omega_d = \omega_d - \eta_\omega \frac{\partial J}{\partial \omega_d} + b_1,$$

$$\omega_{dL} \le \omega_d \le \omega_{dU}$$

(19) $$k_d = k_d - \eta_k \frac{\partial J}{\partial k_d} + b_2, \quad k_{dL} \le k_d \le k_{dU}$$

where,

η_ω, η_k = adaptation constants.

b_1, b_2 = learning biases.

$\omega_{dU}, \omega_{dL}, k_{dU}, k_{dL}$ = Upper and lower limits of variation.

Learning biases are used to recover the original frequencies and gains when the failure is removed. Also, by limiting the frequency and the gain, stability of the reference model is maintained.

5.1.4 An Optimistic Retrospective Critic (ORC)

As a first approximation, it is assumed that the critic's adaptation has progressed in the right direction and that the performance will achieve a stationarity condition in the next time step of interest. This is equivalent to saying that λ_{t+1} *=0*. Using this assumption, the λ's are computed for preceding *n* time steps using equation 10 and the reference model is adapted using the rest of the equations through equation 19. This approach is named here as the Optimistic Retrospective Critic (ORC) to reflect the fact the approach is both optimistic (assuming λ_{t+1} *=0*) and looks at *n* time steps into the past to compute the corrected λ's. The ORC approach can be seen as the bench mark by which the adaptive critic performance can be evaluated.

5.2 Application of the Critic approaches to the C-17

5.2.1 The Problem

The C-17 airplane is a high performance military transport with a quad-redundant Fly-by-Wire Flight Controls System. The flexibility of its control architecture makes it a suitable platform for various types of research in support of safety initiatives and for preliminary investigation of changes to improve production C-17 operation. The C-17 has 22 controllable aerodynamic surfaces and 4 jet engines.

These control surfaces together provide a high level of analytical redundancy in the event of failure. The IFC architecture exploits the analytical redundancy by optimally allocating the surfaces to achieve desired accelerations[16]. In the event of severe damage or failures, the requested response characteristics (both frequency and gain) cannot be attained and hence the reference model needs to be adapted. In the next section, we present results of using the adaptive critic approach for various failure scenarios and for two different critic approaches.