01 Jan 1997

Adaptive Critic Based Neurocontroller for Autolanding of Aircrafts

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

Gaurav Saini

Recommended Citation

S. N. Balakrishnan and G. Saini, "Adaptive Critic Based Neurocontroller for Autolanding of Aircrafts," *Proceedings of the 1997 American Control Conference, 1997*, Institute of Electrical and Electronics Engineers (IEEE), Jan 1997.
The definitive version is available at https://doi.org/10.1109/ACC.1997.609699

Proceedings of the American Control Conference
Albuquerque, New Mexico June 1997
0-7803-3832-4/97/$10.00 © 1997 AACC

ADAPTIVE CRITIC BASED NEUROCONTROLLER FOR AUTOLANDING OF AIRCRAFTS

Gaurav Saini and S.N. Balakrishnan,Contact Person
Department of Mechanical and Aerospace Engineering and Engineering Mechanics
University of Missouri-Rolla. Rolla, MO 65401
Voice: (573) 341-4675, Fax: (573) 341-4607
E-mail: (*gaura@umr.edu*) (*bala@umr.edu*)

Abstract

In this paper, adaptive critic based neural networks have been used to design a controller for a benchmark problem in aircraft autolanding. The adaptive critic control methodology comprises successive adaptations of two neural networks, namely 'action' and 'critic' network (which approximate the Hamiltonian equations associated with optimal control theory) until closed loop optimal control is achieved. The autolanding problem deals with longitudinal dynamics of an aircraft which is to be landed in a specified touchdown region (within acceptable ranges of speed, pitch angle and sink rate) in the presence of wind disturbances and gusts using elevator deflection as the control for glideslope and flare modes. The performance of the neurocontroller is compared to that of a conventional Proportional-Integral-Differential (PID) controller. The results show that the neurocontrollers have good potential for aircraft applications.

1. Introduction

Adaptive critics based neural networks have been used to solve aircraft control problems [1,2]. Adaptive critic method determines optimal control law for a system by successively adapting two neural networks, an action network (which dispenses the control signals) and a critic network (which 'learns' the desired performance index for some function associated with the performance index). In this study, these networks approximate the **Hamiltonian equations** associated with the optimal control theory. The adaptation process starts with a non optimal arbitrarily chosen control and the critic network coerces the action network towards the optimal solution at each successive adaptation. During the adaptations, neither of the networks need any 'information' of a optimal trajectory, only the desired cost needs to be known. Furthermore, this method determines optimal control policy for an entire range of initial conditions and needs no external training as in other form of neurocontrollers.

Aircraft autolanding is a very challenging problem for an adaptive critic based neurocontrol application because (i) an aircraft cannot be trained through crashing as in the case of other problems like inverted pendulum or a robot (ii) conventional linearized controllers cannot emulate pilot responses to emergencies. The autolanding problem deals with linearized aircraft dynamics in the vertical plane; the aircraft has to be landed in a specified touchdown region within acceptable ranges of speed, pitch angle and altitude rate in presence of wind disturbances. The elevator deflection is the only control that guides the aircraft's trajectory for glideslope as well as flare modes. The design of adaptive critic based neurocontroller is presented in the subsequent sections. Also, the optimal flight paths are obtained by solving the LQR formulation using conventional optimal control theory.

2. Aircraft Autolanding

During aircraft landing, the final two phases of a landing trajectory consist of a "glideslope" phase and a "flare" phase. Glideslope is characterized by a linear downward slope; flare by a negative exponential. At approximately 50 feet above the runaway surface, the flare is initiated to elevate the nose of the aircraft, bleed off airspeed, and cause a soft touchdown on the runaway surface. From the flare-initiation point until touchdown, the aircraft follows a control program which decreases both vertical velocity and air speed.

2.1 Linearized Aircraft Equations of Motion

The linearized equations of motion define 2-D incremental aircraft dynamics in the longitudinal / vertical plane. They constitute the bare airframe velocity components, the pitch rate and the angle along with the aircraft position. These equations are developed by assuming that the aircraft is flying in a trimmed condition(i.e., zero translational and rotational accelerations). Small perturbations u, w, q about the mean values are considered and equations of motion are expanded to first order to yield complete longitudinal linearized equations in terms of stability derivatives (X_u, X_w, X_q, Z_u, Z_w, Z_q, M_u, M_w, M_q) and control derivatives (X_E, X_T, Z_E, Z_T, M_E, M_T) [3].

$$\dot{u} = X_u u + \frac{V_{tas}\pi}{180} X_w \alpha + \frac{\pi}{180} X_q q - \frac{\pi}{180} g\theta + X_E \delta_E + X_T \delta_T - X_u u_g - X_w w_g ,$$

$$\dot{\alpha} = \frac{180}{V_{Tas}\pi} Z_u u + Z_w \alpha + \frac{1}{V_{Tas}} (V_{Tas} + Z_q) + \frac{180}{V_{Tas}\pi} (Z_E \delta_E + Z_T \delta_T - Z_u u_g - Z_w w_g) ,$$

$$\dot{q} = \frac{180}{\pi}M_u u + V_{tas}M_w\alpha + M_q q + \frac{180}{\pi}(M_E\delta_E + M_T\delta_T - M_u u_g - M_w w_g),$$

$$\dot{\theta} = q,$$

$$\dot{x} = (V_{Tas}+u)\cos\theta + \frac{V_{Tas}\pi}{180}\alpha\sin\theta \approx V_{Tas}+u,$$

$$\dot{h} = (V_{Tas}+u)\sin\theta - \frac{V_{Tas}\pi}{180}\alpha\cos\theta \approx -\frac{V_{Tas}\pi}{180}\alpha + \frac{V_{Tas}\pi}{180}\theta \quad (2.1)$$

u, α, q, θ are the incremental horizontal velocity (ft/s), angle of attack (deg), pitch rate (deg/s) and pitch angle (deg). x and h are the horizontal range (ft) and altitude (ft), δ_E and δ_T are elevator deflection and throttle settings (control variables), V_{TAS} is the nominal velocity(235.6 ft/s), and u_g and w_g are the wind gust components obtained from Dryden spectra for spatial turbulence distribution.

2.2 Design of Conventional PID Controller

Thrust is used to counter changes in the incremental forward velocity, u, hence from the system model equations described by equations (2.1) the effect of the incremental forward velocity u is neglected. The resulting system model has five state variables namely α, q, θ, x and h. θ_{cmd} is the most important control command which controls the aircraft elevator servomechanism and consequently the pitch up during landing. It can be obtained from Figure 1 by the altitude commands (h_{cmd}) which have different values for glideslope and flare modes as

$$h_{cmd}(t) = -x(t)\tan\gamma_{gs}, x(t_{gs}) = x_{gs} = -\frac{h_{gs}}{\tan\gamma_{gs}}; \gamma_{gs} = glideslope\ angle = 2.75^{o}\ (for glideslope)$$

$$h_{cmd}(t) = \frac{h_f}{V_{TAS}\tan\gamma_{gs}+\dot{h}_{TD}}[V_{TAS}\tan\gamma_{gs}e^{\frac{-(x(t)-x_f)}{\tau_x}} + \dot{h}_{TD}]\ (for\ flare) \quad (2.2)$$

(2.2)

Since the aircraft is flying under reduced power at landing, the throttle and the autothrottle have the minimum effect [3]. Hence for designing the controller only one control variable is considered i.e. δ_E (Equation (2.1)). The horizontal and vertical wind gust components, u_g and w_g can be obtained from the Dryden spectra for spatial turbulence distribution [3]. Once the control from the pitch augmentation system and the gust components are known, the flight of the plane can be simulated for glideslope and flare modes by solving Equation (2.1) using Runge-Kutta method by assuming initial conditions on the states as w(0)=1.0 ft/s, q(0)=0.1 rad/s, θ(0)=0.01 rad, x(0)=-6245 ft, h(0)=300 ft.

3. Adaptive Critic Based Controller for Aircraft Autolanding

3.1 Trajectory Optimization:

Training on single trajectory means that the adaptive critic controller is designed for a constant glideslope angle (in our case 2.75°) for glideslope and flare modes. The autolanding problem needs to be formulated in the Hamiltonian formulation [4], so that the required target equations for action and critic networks are obtained and the required boundary conditions are satisfied. The system equations in Hamiltonian formulation are of the form

$$x_{k+1} = f^k(x_k, u_k) \quad (3.1)$$

Equation (3.1) represents state space representation of a system in discretized form. Note that u_k here represents control at step k. The performance index to be minimized is of the form

$$J_i = \phi(N, x_N) + \sum_{k=i}^{N-1} U^k(x_k, u_k) \quad (3.2)$$

where U^k is the Utility. Next, the Hamiltonian is defined as

$$H^k = U^k + \lambda_{k+1}^T f^k \quad (3.3)$$

Lagrange's multipliers are given by the following Equation (3.4)

$$costate\ equation: \lambda_k = \frac{\delta H^k}{\delta x_k} = \left(\frac{\delta f^k}{\delta x_k}\right)\lambda_{k+1} + \frac{\delta U^k}{\delta x_k}, k=i..N-1 \quad (3.4)$$

$$stationarity\ condition: \frac{\delta H^k}{\delta u_k} = \left(\frac{\delta f^k}{\delta u_k}\right)^T \lambda_{k+1} + \frac{\delta U^k}{\delta u_k} = 0, k=i,..N-1 \quad (3.5)$$

$$boundary\ conditions: \left(\frac{\delta \phi}{\delta x_N} - \lambda_N\right)^T dx_N = 0, \left(\frac{\delta H^i}{\delta x_i}\right)^T dx_i = 0 \quad (3.6)$$

Equation (3.4) in this formulation provides the target for the critic network and the optimality equation (Equation (3.5)) provides the target for the action network. Equation (3.6) supplies the split boundary conditions necessary to solve Equations (3.4)-(3.5). The first condition holds only at final time $k=N$, whereas the second one holds only at initial time $k=i$. In this application, the system starts with a known initial state x_i. So, the second condition holds since $d_i=0$ and there is no constraint on the value of $\delta H/\delta x_i$. Since there is no constraint on the final state x_N, which is typical of a infinite horizon problem, it follows from the first equation that $\lambda_N = \delta\phi/\delta x_N$ i.e. the terminal condition is the value of the final costate λ_N. Also, since all states reach steady state, so $\phi=0$, hence $\lambda_N=0$.

To begin the training procedure, the system equations given by Equation(2.1) are expressed in the desired form ($X(t+1) = AX(t) + Bu(t)$,

Equation 3.1) and hence discretized using a sample time of 1 sec without the effect of wind gust components. $X(t)$ = state vector=$[w(t)\ q(t)\ \theta(t)\ x(t)\ \dot{x}(t)\ h(t)\ \dot{h}(t)]^T$ and $u(t)$ = control= $\delta_E(t)$. The utility $U(x(t))$ is a quadratic function and puts the constraints on the states x and h and the control variable (δ_E) and the only way the networks get information about the commands is through the utility which is defined as

$$U(x(t)) = a_1[h(t) - h_{cmd}(t)]^2 + a_2[\dot{h}(t) - \dot{h}_{cmd}(t)]^2 + a_3\delta_E^2 \; ; \; J = \sum_{t=0}^{t=\infty} U(x(t)) \tag{3.7}$$

where a_1, a_2, a_3 are the respective weightings on the various elements of the utility function and are determined by experimentation. For this problem the values for the various weightings are chosen to be as a_1 = 0.01, a_2 = 1.0 and a_3 = 0.009. The values of h_{cmd} and $\dot{h}_{cmd}$ are obtained for glideslope and flare modes and $\tan\gamma_{gs} = \tan(2.75) = 0.0480$. The cost function is represented by J. A initial arbitrary stabilizing control may be assumed initially as

$$\delta_E^*(t) = -2e-4\left(\sum_{i=1}^{7}(X_i(t))\right) \tag{3.8}$$

Equations (3.4) and (3.5) give the target for the critic network and action network

$$[\lambda_X^*(t)]^T = [\lambda_X(t+1)]^T[A] + \left[\frac{\delta U(x(t))}{\delta X(t)}\right]^T ; \quad \delta_E^*(t) = -\frac{1}{2a_3}(B^T\lambda_x(t+1)) \tag{3.9}$$

$[\lambda_X(t+1)]$ is a 7×1 matrix of the critics at the next time step corresponding to each state and $[\lambda^*_X(t)]$ are the corresponding targets at current time step. Since the utility function in Equation (3.7) is defined in terms of the states, $[\delta U(x(t))/\delta X(t)]$ is available. $[B]$ is the 7×1 control matrix. The training procedure for the neurocontroller is shown in Figure 2.

The action network of $N_{7,2,2,1}$ architecture and random weights is initiated to begin the training. The network inputs the 7 states and outputs the control variable $\delta_E(t)$. As stated earlier, random values of states are not used to train the network. Initial values of states are used as the starting point and the subsequent input values for training are generated using the state equations in Equation (2.1). In order to facilitate effective training of the network, each input(state) needs to be scaled so that all inputs are in a compatible range and no input dominates and overshadows the effect of any other input. Thus, before the states are input into the network, they are scaled down by their respective maximum absolute values, S1. These scaled values of the states, $X_S(t)$ are then input into the network instead of actual values of states, $X(t)$. The target for the initial action network is as given in Equation (3.8). $\delta_E^*(t)$ is found using the actual values of states ($X(t)$) and not the scaled values that are input into the network. Training of the network involves backpropagating the errors between $\delta_E^*(t)$ and $\delta_E(t)$ using the standard backpropagation algorithm. The action network is trained for 10,000 epochs to get a desired level of convergence.

After the action network has been trained for an initial stabilizing control, a new network of architecture $N_{7,2,2,7}$ and random weights is initialized for the critic network. The critic network again inputs scaled values of states $X_S(t)$ and outputs $\lambda_X(t)$. The target for the critic network $\lambda_X^*(t)$ is as given in Equation (3.9). The target equation requires the critics at the next time step, $\lambda_X(t+1)$. Using the state values at current time step ($X(t)$), the plant model and the action network are used to find the states at the next time step, $X(t+1)$. These states are then scaled again and the scaled values of states $X_S(t+1)$ are used to find the critic values at next time step, $\lambda_X(t+1)$ which are used in the target critic equation. $\delta U(x(t))/\delta X(t)$ is calculated using the actual states at every time step. Once the elements of the target critic equations are known, $\lambda^*_X(t)$ is found by implementing Equation (3.9). Before backpropagating the errors between $\lambda_X(t)$ and $\lambda^*_X(t)$ to train the critic network, $\lambda^*_X(t)$ is scaled down by the respective maximum absolute values, S2. The critic network is trained for all the points in the trajectory for 10,000 epochs using backpropagation. This completes the training of first critic.

After the first critic network converges, the action network is initiated again, not with the random weights but with the weights of initial control. The network inputs scaled values of states at every time step and outputs the control variable $\delta_E(t)$ and is trained for a target control, $\delta_E^*(t)$ given in Equation (3.9).

$\lambda_X(t+1)$ are the critic values at the next time step and are obtained from the trained critic network. Since the target critic values, $\lambda^*_X(t)$ were scaled down by a factor S2 during the training of critic network, the $\lambda_X(t+1)$ values are scaled up by the same scaling factor, S2 before they are used to calculate the target control $\delta_E^*(t)$. This target control is in turn scaled down by a factor S3 before the error is backpropagated into the action network. This convergence of the action network completes the training of the first action network. After training of the first action, the critic is again initiated, not with random weights but with the weights of the first critic. The process of training the subsequent critic and action networks is exactly similar to the training of first critic and action networks. At any time during the training process, the weights of previous action or critic networks are taken as starting weights. This process of successively adapting the action and critic network continues until both networks converge which takes place in six adaptive critic cycles.

Figure 3 shows the structure of the adaptive critic-based controller for a single trajectory. $X(t)$ is a vector of all the states mentioned in Equation (2.1) and they are scaled by their respective maximum absolute values (S1) before they are input to action or critic networks. Scaled values of states are input to the action network and the action network outputs the control, $\delta_E(t)$. Target control $\delta_E^*(t)$ is scaled down by a scaling factor S3 to facilitate efficient training of the action network.

The states at the next time step $X(t+1)$ are generated using the known aircraft model and the action network (the control is scaled up by factor S3). Thus, scaled values of states at time$(t+1)$ and t are input into the critic network successively which outputs $\lambda_X(t+1)$ and $\lambda_X(t)$, correspondingly. The target critics are calculated using Equation (3.9) and they are scaled down by a factor S2 before backpropagating the error in the critic network. The critic values are again scaled up by the factor S2 for calculating the target control (not shown in the figure).

3.2 Optimal Control for Aircraft Autolanding

Optimal control theory provides the formulation of a discrete-time linear quadratic regulator for linear systems with quadratic performance indices for the free final state (infinite horizon) class of problems which leads to closed loop control [4]. The closed loop optimal control problem is again formulated as a two point boundary value

problem as described in section III before. The quadratic cost function is of the form

$$J_0 = \frac{1}{2}\sum_{k=0}^{\infty}(x_k^T Q x_k + u_k^T R u_k) \qquad (3.10)$$

For our problem, the plant (A, B) and cost-weighting (Q, R) matrices are time invariant. The cost weighting matrices can be obtained from Equation (3.7). This formulation demands that the constant nominal velocity, V_{TAS} be introduced as a state, which triggers uncontrollability in the system. To obviate this, a fictitious control is introduced in the system equations which controls this state, and to minimize its effect in the system dynamics it is weighed very heavily in the cost weighting matrix, R. Once these matrices are known, steady state optimal gains can be obtained from the eigenvectors of the Hamiltonian matrix [4] which can be used to find the optimal trajectories.

4. Results and Conclusions

Numerical results from our test cases are presented in Figures 3-8. The pitch angle and pitch rate histories are presented in Figures 3 and 4. After the transient induced due to initiation of glide slope, the aircraft attitude is almost constant. The second phase of disturbances around 22 seconds is due to the initiation of flare. From Figure 5, it can be observed that the neurocontroller makes the aircraft follow the commanded trajectory quite well. The altitude rate plot (Figure 6) shows the reduction in sink rate when flare is initiated. The behavior of the aircraft under gust are presented in Figures 7 and 8. It can be observed that the neurocontroller is much smoother than the PID controller. It is clear from the numerical results that a neurocontroller is very capable at control of aircraft.

Note that we have used the glideslope mode in Figures 7 and 8. Switching to flare mode is straightforward. Furthermore, it is found that to solve the autolanding problem using the conventional linear quadratic regulator (LQR) method, the formulation needs to be more rigid (only quadratic cost functions). Use of neurocontrollers for non-quadratic performance index is exactly the same as for quadratic; however, any other formulation requires a lot of assumptions/approximations to obtain a feedback law. This research was supported by NSF (National Science Foundation), Dr. Paul Werbos is the program manager.

5. References

[1] Balakrishnan, S., and V. Biega,"Adaptive-Critic based Neural Networks for Aircraft Control, "*AIAA Journal of Guidance, Control and Dynamics*, July 1996.

[2] Prokhrov, D., R. Santiago, and D. Wunsch, "Adaptive Critic Designs: A case study for Neurocontrol," *Neural Networks*, vol 5, 1995.

[3] Jorgensen, C., and C. Schley, "A Neural Network Baseline Problem for Control of Aircraft Flare and Touchdown," *Neural Networks for Control* (W.T. Miller, R.S. Sutton, and P.J. Werbos, eds.) ch. 17, pp. 403-425, MIT Press, 1990.

[4] Lewis, F.L., "Optimal control of discrete-time systems" *Optimal Control* (John Wiley & Sons Inc.) ch.2, pp 25-143, 1986.

[5] Werbos, P.J., "Optimization Methods for Brain-like Intelligent Control," *Proceedings of the 34th Conference on Decision and Control*,", Dec. '95, pp.579-584

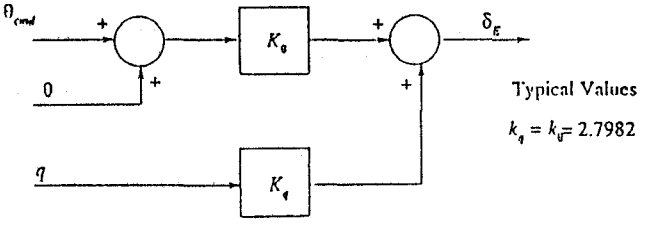
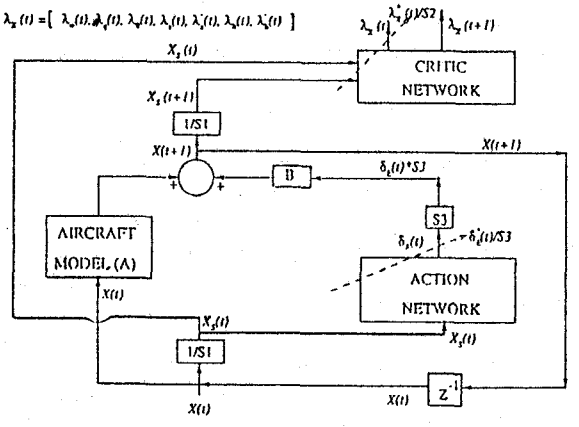


Figure 1. Pitch stability augmentation system



$X(t)$ = [$w(t)$, $q(t)$, $\theta(t)$, $x(t)$, $\dot{x}(t)$, $h(t)$, $\dot{h}(t)$]
B = Control Matrix

Figure 2. Structure of adaptive critic based controller for autolanding

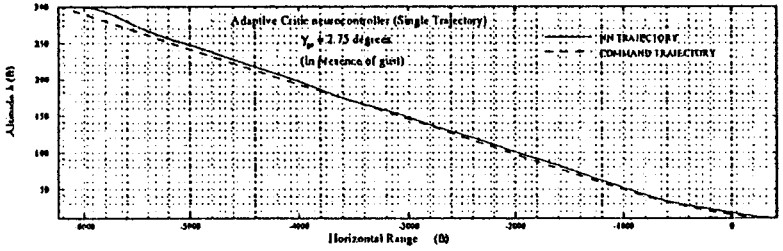


Figure 7. Flight path (h v/s x) with neurocontroller (single trajectory) with gust

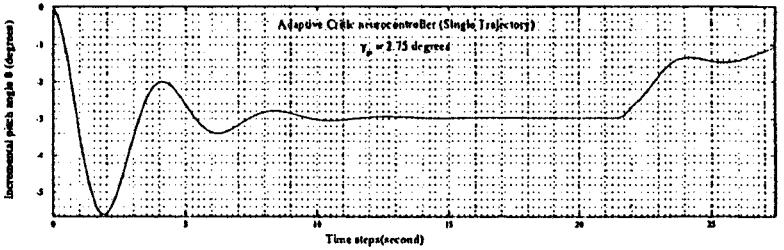


Figure 3. Response of state θ v/s time with neurocontroller (single trajectory)

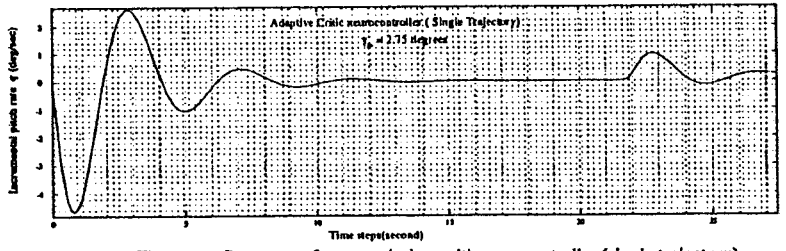


Figure 4. Response of state q v/s time with neurocontroller (single trajectory)

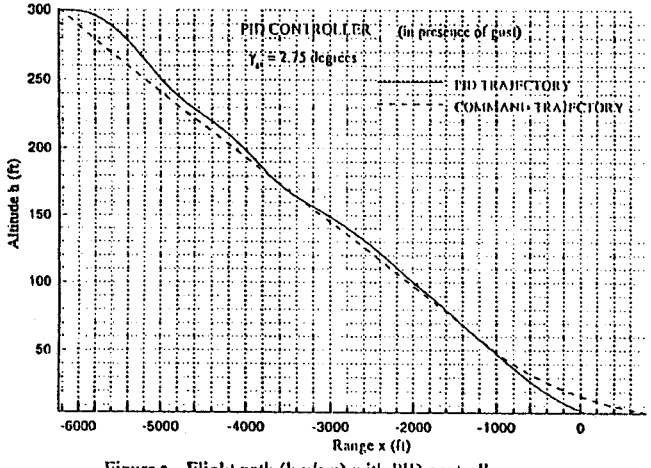


Figure 8. Flight path (h v/s x) with PID controller in presence of gust

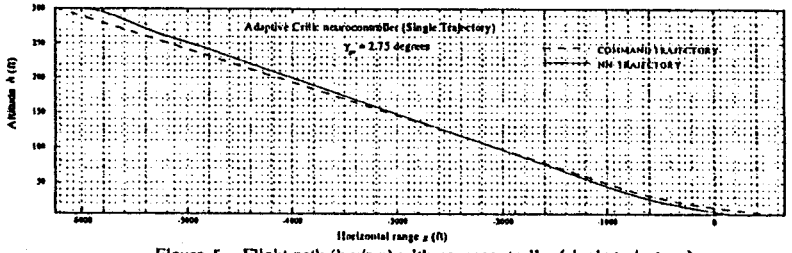


Figure 5. Flight path (h v/s x) with neurocontroller (single trajectory)

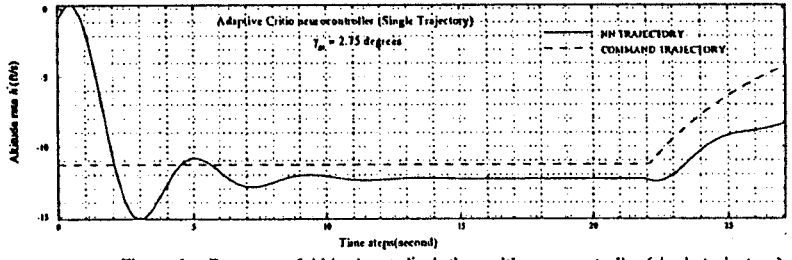


Figure 6. Response of Altitude rate ḣ v/s time with neurocontroller (single trajectory)