

01 Jan 1999

Adaptive Critic Based Neural Networks for Control-Constrained Agile Missile Control

Dongchen Han

S. N. Balakrishnan

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

Follow this and additional works at: https://scholarsmine.mst.edu/mec_aereng_facwork



Part of the [Aerospace Engineering Commons](#), and the [Mechanical Engineering Commons](#)

Recommended Citation

D. Han and S. N. Balakrishnan, "Adaptive Critic Based Neural Networks for Control-Constrained Agile Missile Control," *Proceedings of the 1999 American Control Conference, 1999*, Institute of Electrical and Electronics Engineers (IEEE), Jan 1999.

The definitive version is available at <https://doi.org/10.1109/ACC.1999.786536>

This Article - Conference proceedings is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Mechanical and Aerospace Engineering Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. This work is protected by U. S. Copyright Law. Unauthorized use including reproduction for redistribution requires the permission of the copyright holder. For more information, please contact scholarsmine@mst.edu.

ADAPTIVE CRITIC BASED NEURAL NETWORKS FOR CONTROL-CONSTRAINED AGILE MISSILE CONTROL

Dongchen Han and S. N. Balakrishnan
University of Missouri-Rolla
Rolla, MO 65409-0050

ABSTRACT

In this study we investigate the use of an 'adaptive critic' based controller to steer an agile missile with a constraint on the angle of attack from various initial Mach numbers to a given final Mach number in minimum time while completely reversing its flightpath angle. We use neural networks with a two-network structure called 'adaptive critic' in this study to carry out the optimization process. This structure obtains an optimal controller through solving Hamiltonian equations. This approach needs no external training; each network along with the optimality equations generates the output for the other network. When the outputs are mutually consistent, the controller output is OPTIMAL. Though the networks are trained off-line, the resulting control is a feedback control.

1. INTRODUCTION

In order to explore and extend the range of operations of air-to-air missiles, there have been studies in recent years with a completely different concept. That is launch the missile as usual from the aircraft; however, the missile should be able to intercept a target in the REAR hemisphere. The best emerging alternative to execute this task is to use the aerodynamics and thrust to turn around the initial flight path angle of zero to a final flight path angle of 180 degrees. (Every scenario can be considered as a subset of this set of extremes in flightpath angle.) Furthermore, the missile is constrained to fly with limited angle of attack. In the design as well as later phases of such a missile, there is a need to develop analysis tools to study such a minimum time problem under various initial conditions. This problem falls under a class called 'free final time control-constrained' problems in calculus of variations (optimal control) which for an envelope of initial conditions is difficult to solve. Currently there is no unified mathematical formalism under which a controller can be designed for nonlinear systems. Available solutions for nonlinear controllers are highly problem oriented. Consequently, we propose a formulation which: 1) solves a nonlinear control problem directly without any approximation to the system model (in the absence of a good model this approach can synthesize a nonlinear model of the states), 2) yield a control law in a feedback form as a function of the current states, and 3) maintain the same structure regardless of the type or problem (handles linear problems as well). Such a formulation is afforded by the field of neural networks.

Several authors have used neural networks to "optimally" solve nonlinear systems [Hunt [2], White and Sofge [3]]. Almost all these studies fall within four categories: 1) supervised control, 2) direct inverse control, 3) neural adaptive control, and 4) backpropagation through time [7]. A fifth and rarely studied class of controller has the most interesting structure. It is called an Adaptive Critic

Architecture [3,4,5,6,7], Fig. 1. The reason for choosing this structure for formulating the optimal control problems are that this approach needs NO external training as in other forms of neurocontrollers, this is not an open loop optimal controller but a feedback controller, and it preserves the same structure regardless of the problem (linear or nonlinear). Balakrishnan and Biega [4] have shown the usefulness of this architecture for infinite finite-time linear problems.

The method discussed in this study determines an optimal control law for a nonlinear system by successively adapting two networks, an action network and a critic network. This method determines the control law for an entire range of initial conditions [4]. It simultaneously determines and adapts the neural networks to the optimal control policy for both linear and nonlinear systems.

We have solved the unconstrained problem by this method [8]. In this paper we study the case with control constraints.

2. PROBLEM FORMULATION AND SOLUTION DEVELOPMENT

2.1 Mathematical Formulation

Through the neural network methodology presented in this study, we will be able to solve a class of optimal control problems.

The system model is given by

$$\mathbf{x}(i+1) = \mathbf{f}_i(\mathbf{x}(i), \mathbf{u}(i)) \quad (1)$$

where $\mathbf{f}_i(\cdot)$ can be either linear or nonlinear. Find a optimal control $\mathbf{u}(i)$ that

$$\min J = \phi[\mathbf{x}(N)] + \sum_{i=0}^{N-1} L_i[\mathbf{x}(i), \mathbf{u}(i)]. \quad (2)$$

Subject to a control inequality constraint:

$$C[\mathbf{u}(i)] \leq 0 \quad (3)$$

In Eq. (2), $L_i(\cdot)$ can be a linear or nonlinear function of the states and/or control and $\phi(\cdot)$ can be a linear or nonlinear function of terminal states. $L_i(\cdot)$ is also known as a utility function; i indicates the stage.

The optimal control problem can be formulated in terms of Hamiltonian [1] where the Hamiltonian, H_i , is given by

$$H_i = L_i(\mathbf{x}(i), \mathbf{u}(i)) + \lambda^T(i+1) \mathbf{f}_i(\mathbf{x}(i), \mathbf{u}(i)) + \mu^T(i+1) C[\mathbf{u}(i)] \quad (4)$$

The propagation equations for the lagrange multiplier, $i=0, 1, \dots, N-1$, are given by

$$\lambda(i) = (\partial \mathbf{f}_i / \partial \mathbf{x}(i))^T \lambda(i+1) + (\partial L_i / \partial \mathbf{x}(i))^T \quad (5)$$

with boundary condition on λ as

$$\lambda(N) = (\partial\phi(x(N))/\partial x(N))^T \quad (6)$$

The optimality condition is

$$\partial H_i / \partial u(i) = 0, \quad i = 0, 1, \dots, N-1 \quad (7)$$

$$\text{i.e.,} \quad \frac{\partial H_i}{\partial u(i)} = \frac{\partial L_i}{\partial u(i)} + \lambda^T(i+1) \frac{\partial f_i}{\partial u(i)} + \mu^T(i+1) \frac{\partial C[u(i)]}{\partial u(i)} \quad (8)$$

With the additional requirement that:

$$\mu(i) \begin{cases} \geq 0 & C=0 \\ =0 & C<0 \end{cases} \quad (9)$$

For problem with control variable inequality constraints, the following equations hold at the junction between constrained arc and unconstrained arc [1]:

$$\begin{aligned} \lambda(i^-) &= \lambda(i^+) \\ H(i^-) &= H(i^+) \\ H_u(i^-) &= H_u(i^+) \end{aligned} \quad (10)$$

i.e the control inequality constraint will not form a corner, that is the λ , H , u , μ are continuous across the junction points between the unconstrained control arc and constrained control arc. So the control inequality constraint problem is different from unconstrained problem only in that it needs to get $\mu(i)$.

For $c < 0$ we have $\mu(i) = 0$ and equation (8) determine $u(i)$. For $C = 0$ the state equation (2) determine $u(i)$ then equ.(8) calculate $\mu(i)$.

2.2 General Procedure for Adaptive Critic Solution

For finite time (or finite-horizon) problems, solution with neural networks evolves in two stages:

Synthesis of the Last Network

- 1) Note that $\lambda(N) = \left(\frac{\partial\phi(x(N))}{\partial x(N)} \right)^T$. For various random values of $x(N)$, λ_N can be calculated.
- 2) Use the state-propagation Eq. (1) and optimality condition in Eq. (7) to calculate u_{N-1} for various x_{N-1} by randomly selecting $x(N)$ and the corresponding λ_N from step 1.
- 3) With u_{N-1} and λ_N , calculate λ_{N-1} for various x_{N-1} by using the costate propagation equation (5).
- 4) Train two neural networks. For different values of x_{N-1} , the u_{N-1} network outputs u_{N-1} and the λ_{N-1} network outputs λ_{N-1} . We have optimal control and costates for various values of the state at stage (N-1) now.

Other Networks

- 1) Assume different values of the states at stage (N-2), x_{N-2} , and use a random network (or initialized with u_{N-1} network) called u_{N-2} network to output u_{N-2} . Use u_{N-2} and x_{N-2} in the state propagation equation to get x_{N-1} . Input x_{N-1} to the λ_{N-1} network to obtain λ_{N-1} . Use x_{N-2} and λ_{N-1}

in the optimality condition in Eq. (7) to get target u_{N-2} . Use this to correct the u_{N-2} network. Continue this process till the network weights show little changes. This u_{N-2} network yields optimal u_{N-2} .

- 2) Using random x_{N-2} , output the control u_{N-2} from the u_{N-2} network. Use these x_{N-2} and u_{N-2} to get x_{N-1} and input x_{N-1} to generate λ_{N-1} . Use x_{N-2} , u_{N-2} and λ_{N-1} to obtain optimal λ_{N-2} . Train a λ_{N-2} network with x_{N-2} as input and obtain optimal λ_{N-2} as output.
- 3) Repeat the last two steps with $k = N-1, N-2, \dots, 0$ until we get u_0 .

A schematic of the network development is presented in Fig. 2.

3. SYSTEM MODEL IN A VERTICAL PLANE

3.1 The Reformulation of System Model

The motion equations of a agile missile in a vertical plane are presented in this section. The minimum-time optimization problem is presented, the difficulties are pointed out, and a reformulation is made with the flightpath angle as the independent variable.

The non-dimensional equations of motion in a vertical plane are:

$$M' = -S_w M^2 C_D - \sin \gamma + T_w \cos \alpha \quad (11)$$

$$\gamma' = \frac{1}{M} \left[S_w M^2 C_L + T_w \sin \alpha - \cos \gamma \right] \quad (12)$$

$$X_E' = M \cos \gamma \quad (13)$$

$$Z_E' = -M \sin \gamma \quad (14)$$

where prime denotes differentiation with respect to the nondimensional time, τ .

The nondimensional parameters used in Eqn. (11) - (14) are:

$$\tau = g/at ; T_w = T/mg ; S_w = \rho a^2 S/2mg$$

$$M = V/a ; X_E = \frac{g}{a^2} x ; Z_E = \frac{g}{a^2} z$$

In these equations, M is the flight Mach number, γ , the flightpath angle, α , the aerodynamic angle of attack, x , the horizontal range, z , the negative of altitude (pointing down), T , the solid rocket thrust, m , the mass of the missile, S , the reference aerodynamic area, V , the speed of the missile, C_L , the lift coefficient, C_D , the drag coefficient, g , the acceleration due to gravity, a , the speed of sound, ρ , the atmospheric density, and t is the flight time. Note that C_D and C_L are functions of angle of attack and flight Mach number.

Objective:

The objective of the minimization process is to find the control (angle-of-attack) history to minimize the time taken by the missile to reverse its flightpath angle completely with the limited angle of attack while the Mach number changes from an envelope of initial Mach numbers to a given final Mach number of 0.8.

Mathematically, this problem is stated as to find the control minimizing J , the cost function where

$$J = \int_0^{t_f} dt \quad (15)$$

with the constraints $\gamma(0) = 0$ deg., $M(0) = M_0$ given, $\gamma(t_f) = 180$ deg. and $M(t_f) = 0.8$. This constrained optimization problem comes under the class of 'free final time' problems in calculus of variations and is difficult to solve. No general solution exists which generates optimal paths for flexible initial conditions.

We seek to provide such solutions using adaptive critic-based neural networks. In order to facilitate the solution using neural networks, the equations of motion are reformulated using the flightpath angle as the independent variable. This enables us to have a fixed final condition as opposed to the 'free final time'. The transformed dynamic equations are:

$$\frac{dM}{d\gamma} = \frac{(-S_w M^2 C_D - \sin \gamma + T_w \alpha) M}{S_w M^2 C_L - \cos \gamma + T_w \sin \alpha} \quad (16)$$

and transformed cost function:

$$J = \int_0^{\pi} \frac{aM}{g(S_w M^2 C_L \cos \gamma + T_w \sin \alpha)} d\gamma \quad (17)$$

subject to the control variable inequality constraint:

$$\alpha \leq \alpha^* \quad (18)$$

(here $\alpha^* = 120^\circ$)

$$C[u(i)] = \alpha - \alpha^* \leq 0 \quad (19)$$

The corresponding Hamiltonian equation is: (continuous form)

$$H = L + \lambda f + \mu C \quad (20)$$

Written in discrete form as:

$$\begin{aligned} H_k &= \left(\frac{dt}{d\gamma} \right)_k \cdot \delta \gamma + \lambda_{k+1} \cdot M_{k+1} + \mu_{k+1} \cdot C[u(i+1)] \\ &= \frac{aM_k \cdot \delta \gamma}{g(S_w M_k^2 C_{Lk} - \cos \gamma_k + T_{wk} \sin \alpha_k)} \quad (21) \\ &+ \lambda_{k+1} \left[M_k + \frac{(-S_w M_k C_{Dk} - \sin \gamma_k + T_{wk} \cos \alpha_k) M_k \cdot \delta \gamma}{S_w M_k^2 C_{Lk} - \cos \gamma_k + T_{wk} \sin \alpha_k} \right] + \mu_{k+1} \cdot \alpha_k \end{aligned}$$

Let

$$\text{denk} = S_w M_k^2 C_{Lk} - \cos \gamma_k + T_{wk} \sin \alpha_k \quad (22)$$

$$\frac{\partial \text{denk}}{\partial \alpha_k} = S_w M_k^2 \frac{\partial C_{Lk}}{\partial \alpha_k} + T_{wk} \cos \alpha_k \quad (23)$$

$$\frac{\partial \text{denk}}{\partial M_k} = 2S_w M_k C_{Lk} + S_w M_k^2 \frac{\partial C_{Lk}}{\partial M_k} \quad (24)$$

the costate equation is:

$$\begin{aligned} \frac{\partial H_k}{\partial M_k} = \lambda_k &= \frac{a \cdot \delta \gamma}{g \cdot \text{denk}} - \frac{a M_k \delta \gamma}{g \cdot \text{denk}^2} \cdot \frac{\partial \text{denk}}{\partial M_k} \\ &+ \lambda_{k+1} \cdot \delta \gamma \left(\frac{-3S_w M_k^2 C_{Dk} - S_w M_k^3 \frac{\partial C_{Dk}}{\partial M_k} - \sin \gamma_k + T_{wk} \cos \alpha_k}{\text{denk}} \right) \quad (25) \\ &\lambda_{k+1} + \lambda_{k+1} \cdot \delta \gamma \left(\frac{S_w M_k^2 C_{Dk} + \sin \gamma_k - T_{wk} \cos \alpha_k}{\text{denk}^2} \right) \cdot M_k \cdot \frac{\partial \text{denk}}{\partial M_k} \end{aligned}$$

Optimality condition is obtained as

$$\frac{\partial H_k}{\partial \alpha_k} = 0 \quad (26)$$

In an expanded form, this is

$$\begin{aligned} \frac{a}{g} \cdot \frac{\partial \text{denk}}{\partial \alpha_k} + \lambda_{k+1} \cdot \left(S_w M_k^2 \frac{\partial C_{Dk}}{\partial \alpha_k} + T_{wk} \sin \alpha_k \right) \cdot \text{denk} \\ + \lambda_{k+1} \cdot \left(-S_w M_k^2 C_{Dk} - \sin \lambda_k + T_{wk} \cos \alpha_k \right) \cdot \frac{\partial \text{denk}}{\partial \alpha_k} + \mu_{k+1} = 0 \quad (27) \end{aligned}$$

3.2 Development of Neural Network Solutions on the Constraint Boundary

The procedure to develop neural network solution is similar to [8]. There are a few points that need to be noted, however.

- 1) We have developed a neural network solution corresponding to the unconstrained state variable problem. For control inequality constraint, parts of solution can still be used. Since the optimal solution from boundary segment to final state is one of the optimal trajectories of unconstrained problems.
- 2) We get the solution using backward procedure as the unconstrained problem. Prior the control limit is met, all the trained networks are still the optimal solution. We continue the procedure until the limit control is exceeded, then we just let the control target equal the limit value. If the state through this control falling the previous one trained network scope, then this is the optimal control, otherwise choose new state scope until all the state falling into the previous trained network scope. Then this network is trained optimally. Thus actually, we needn't calculate the μ value since it doesn't affect the procedure.

4. USE OF NETWORKS IN REAL-TIME AS FEEDBACK CONDITIONS

Assume any M_0 [within the trained range]. Use α_0 neural network to find optimal α and integrate until γ_1 for α_1 neural network is reached; use the M_1 values to find α_1 , from the α_1 neural network and integrate until γ_2 is reached, and so on, until γ_f is reached.

Note that the forward integration is done in terms of time. As a result, even though the network synthesis is done off-line, the control is a feedback process based on current

states.

5. NUMERICAL RESULTS

Tables of aerodynamic data of C_L and C_D variations with Mach numbers and angle of attack were provided by the Air Force. Neural network solutions were developed as described in earlier sections.

The control inequality constraint is chosen such that the angle of attack,

$$\alpha \leq 120 \text{ deg}$$

Before meet the control limit, we adopt all the trained network for unconstrained problem since all these are optimal solution. We again get 37 networks to implement this optimal process. Figures 3-7 are one set of optimal neural network solutions. Note that all these results are forward integration in terms of time. The real advantage of using the adaptive critic approach is clear from Figure 8. For each trajectory with initial Mach number varying from 0.6 to 0.8, the final Mach number is 0.8. That is, the same cascade of neurocontroller is used to generate optimal control for an envelope of initial conditions. To compare the control inequality constraint with the unconstrained problem, we also plot the angle of attack trajectory vs the flight path angle for both the unconstrained and control constrained problem, see Fig. 9 and Fig. 10. From these results we could see that the control limit is met during relative short period, and its effect is approximate to unconstrained control. From the Mach number trajectory we can see that the Mach number is slight high than its corresponding part without constraint. This is expected since smaller angle of attack result in higher speed.

6. CONCLUSIONS

An adaptive critic-based neural network solution for a 'bounded control, free final time' problem associated with agile missile control has been solved. The neural network controllers are able to provide (near) optimal control to the missile from an envelope of initial Mach numbers to a fixed final Mach number of 0.8 in minimum time. An added advantage in using these neurocontrollers is that they provide minimum time solutions even when we change the initial flight path angle from zero to any non zero (positive) value. To our knowledge, there has been no one tool (other than dynamic programming) which provides such solutions.

REFERENCES

[1] A. E. Bryson and Y. Ho, *Applied Optimal Control*, Hemisphere Publishing Co., 1975, pp. 128-211.
 [2] K. J. Hunt, "Neural Networks for Controller Systems, A Survey," *Automatica*, Vol. 28, No. 6, 1992, pp. 1083-1112.
 [3] D. A. White and D. Sofge, *Handbook of Intelligent Control*, Van Nostrand Reinhold, 1992, Ch. 3,5,8,12,13.
 [4] S. N. Balakrishnan and V. Biega, "Adaptive Critic Based Neural Networks for Aircraft Optimal Control," *Journal of Guidance, Control, and Dynamics*, Vol. 19, No. 4, July - August 1996, pp. 893-898.
 [5] C. W. Anderson, "Learning to Control an Inverted Pendulum using Neural Networks," *IEEE Control*

Systems Magazine, April 1898, Vol. 9, pp. 31-37.

[6] A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike Adaptive Elements That Can Solve Difficult Learning Control Problems," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-13, 1983, pp. 834-846.
 [7] P. Werbos, "Neurocontrol and Supervised Learning; An Overview and Evaluation," *Handbook of Intelligent Control*, Van Nostrand Reinhold, 1992.
 [8] D. Han and S.N. Balakrishnan, "Adaptive Critic Based Neural Networks for Agile Missile Control," AIAA Guidance, Navigation, and Control Conference and Exhibit, Boston, Massachusetts, August 10-12, 1998, pp. 1803-1812.

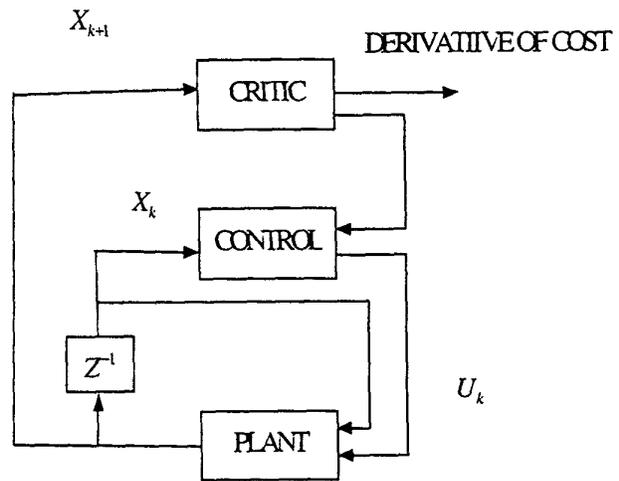


Fig. 1 Schematic of Adaptive Critic Formulation

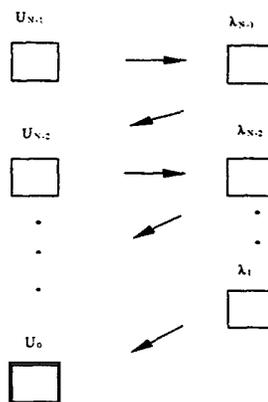


Fig. 2 Schematic of Successive Adaptive Critic Synthesis

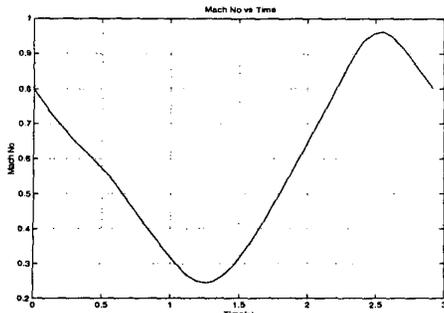


Fig. 3 Mach Number vs Time ($M_0=0.8$)

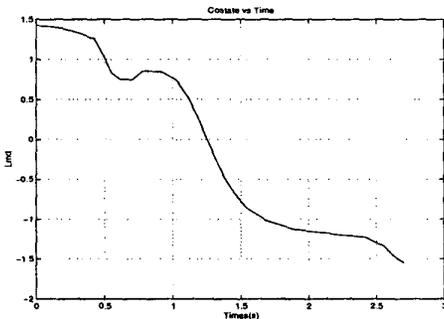


Fig. 4 Co-state vs Time ($M_0=0.8$)

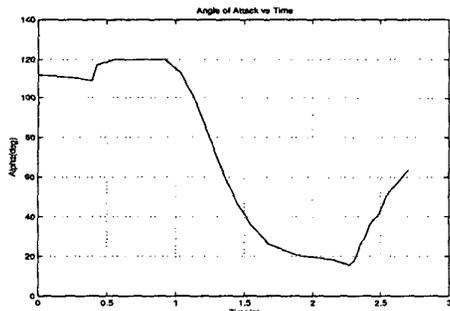


Fig. 5 Angle of Attack vs Time ($M_0=0.8$)

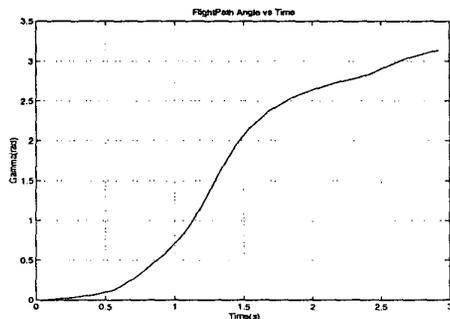


Fig. 6 Flightpath Angle vs Time ($M_0=0.8$)

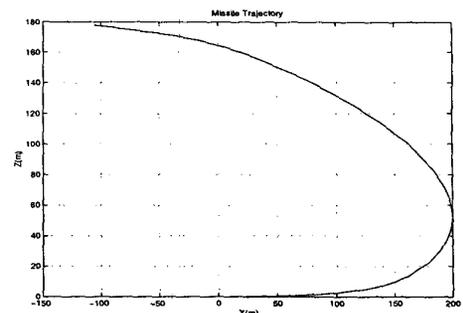


Fig. 7 Missile Trajectory in Vertical Plane ($M_0=0.8$)

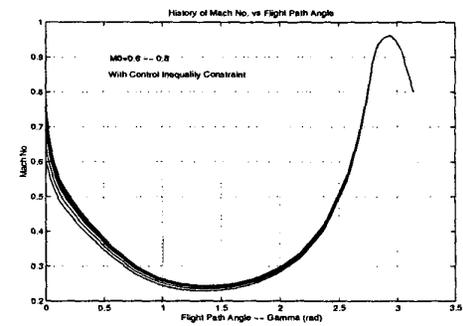


Fig. 8 History of Mach No. vs Flightpath Angle with Different Initial Mach No.

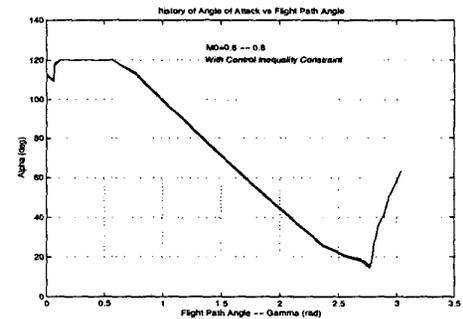


Fig. 9 History of Angle of Attack with Value Limitation

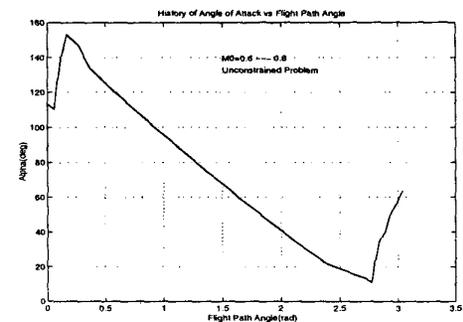


Fig. 10 History of Angle of Attack without Value Limitation