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: http://scholarsmine.mst.edu/faculty_work
Part of the Aerospace Engineering Commons, and the Mechanical Engineering Commons

Recommended Citation
http://scholarsmine.mst.edu/faculty_work/1987

This Article is brought to you for free and open access by Scholars' Mine. It has been accepted for inclusion in Faculty Research & Creative Works by an authorized administrator of Scholars' Mine. For more information, please contact weaverjr@mst.edu.
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 every scenario can be considered as a subset of this set of extremes in flightpath angle.

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.

2. PROBLEM FORMULATION AND SOLUTION DEVELOPMENT

2.1 Mathematical Formulation

The system model is given by

\[ x(i+1) = f_i(x(i), u(i)) \]  

where \( f_i() \) can be either linear or nonlinear. Find a optimal control \( u(i) \) that

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

Subject to a control inequality constraint:

\[ C[u(i)] \leq 0 \]  

In Eq. (2), \( L_i() \) 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. \( L_i() \) 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, \( H_o \), is given by

\[ H_o = L_i[x(i), u(i)] + \lambda^T(i+1)[f_i(x(i)), u(i)] + \mu^T(i+1)C[u(i)] \]  

The propagation equations for the lagrange multiplier, \( i = 0, 1, \ldots, N-1 \), are given by

\[ \lambda(i) = (\partial f_i / \partial x(i))^T \lambda(i+1) + (\partial L_i / \partial x(i))^T \]
with boundary condition on \( \lambda \) as

\[
\lambda(N) = \left( \frac{\partial\phi(x(N))}{\partial x(N)} \right)^T \tag{6}
\]

The optimality condition is

\[
\frac{\partial H_i}{\partial u(i)} = 0, \quad i = 0, 1, \ldots, N-1 \tag{7}
\]

i.e.,

\[
\frac{\partial H_i}{\partial u(i)} = \frac{\partial \lambda}{\partial u(i)}(i+1) - \lambda(i+1) \frac{\partial f}{\partial u(i)} + \lambda(i+1) \frac{\partial C[u(i)]}{\partial u(i)} \tag{8}
\]

With the additional requirement that:

\[
\mu(i) \begin{cases} 
  \geq 0 & C=0 \\
  = 0 & C<0 
\end{cases} \tag{9}
\]

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

\[
\lambda^+(i) = \lambda^-(i) \]

\[
H^+(i) = H^-(i) \]

\[
H_u^+(i) = H_u^-(i) \tag{10}
\]

i.e., the control inequality constraint will not form a corner, that is the \( \lambda, H, u, \mu \) 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 \( \mu < 0 \) 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 \( x(N) \), \( \lambda_N \) 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 \( \lambda_N \) from step 1.

3) With \( u_{N-1} \) and \( \lambda_N \), calculate \( \lambda_{N-1} \) 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 \( u_N \) and the \( \lambda_{N-1} \) network outputs \( \lambda_{N-1} \). 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 \( u_{N-2} \) network) called \( u_{N-2} \) network to output \( u_N \). Use \( u_N \) and \( x_{N-2} \) in the state propagation equation to get \( x_{N-1} \). Input \( x_{N-1} \) to the \( \lambda_{N-1} \) network to obtain \( \lambda_{N-1} \). Use \( x_{N-2} \) and \( \lambda_{N-1} \) 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 \( \lambda_{N-1} \). Use \( x_{N-2} \), \( u_{N-2} \) and \( \lambda_{N-1} \) to obtain optimal \( \lambda_{N-2} \). Train a \( \lambda_{N-2} \) network with \( x_{N-2} \) as input and output optimal \( \lambda_{N-2} \) as output.

3) Repeat the last two steps with \( k = N-1, N-2, \ldots \) 0 until we get \( u_N \).

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 an 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:

\[
M' = -S_mM^2C_D \sin \gamma + T_w \cos \alpha \tag{11}
\]

\[
\gamma' = \frac{1}{M} \left[ S_mM^2C_L + T_w \sin \alpha - \cos \gamma \right] \tag{12}
\]

\[
X'_e = M \cos \gamma \tag{13}
\]

\[
Z'_e = -M \sin \gamma \tag{14}
\]

where prime denotes differentiation with respect to the nondimensional time, \( \tau \).

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

\[
\tau = g/\alpha ; \ T_w = T/mg ; \ S_w = \rho a^2 S/2mg
\]

\[
M = V/a ; \ X_e = \frac{E}{a^2} x ; \ Z_e = \frac{E}{a^2} z
\]

In these equations, \( M \) is the flight Mach number, \( \gamma \), the flightpath angle, \( \alpha \), 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, \( C_D \), the lift coefficient, \( C_L \), the drag coefficient, \( g \), the acceleration due to gravity, \( a \), the speed of sound, \( \rho \), 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
\[ J = \int_0^t \frac{dt}{J(t)} \]  

with the constraints \( \gamma(0) = 0 \text{ deg.}, M(0) \equiv \text{given}, \gamma(t_f) = 180 \text{ deg.} \) 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:

\[ \frac{dM}{dy} = \frac{(-SM^2CL_0 \sin \gamma + T_w \alpha)M}{S \theta^2 M^2CL - \cos \gamma + T_w \sin \alpha} \]  

and transformed cost function:

\[ J = \int_0^t \frac{aM}{g(S \theta^2 M^2CL_0 \cos \gamma + T_w \sin \alpha)} dy \]

subject to the control variable inequality constraint:

\[ \alpha \leq \alpha^* \]

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

\[ C[u(t)] = \alpha^* - \alpha^* < 0 \]  

The corresponding Hamiltonian equation is: (continuous form)

\[ H = L + \lambda \gamma + \mu \gamma \]  

Written in discrete form as:

\[ H_k = \left( \frac{dt}{dy} \right)_k \cdot \lambda_k \cdot \gamma_k + \mu_k \cdot \gamma_k \cdot C[u(t+1)] \]

\[ = \frac{aM_k \cdot \gamma_k}{g(S \theta^2 M^2CL_k - \cos \gamma_k + T_w \sin \gamma_k)} \]  

\[ + \lambda_k \cdot \left( M_k \cdot \frac{(-SM^2CL_0 \sin \gamma_k + T_w \sin \gamma_k)M_k \cdot \gamma_k}{S \theta^2 M^2CL_k - \cos \gamma_k + T_w \sin \gamma_k} \right) + \mu_k \cdot \alpha_k \]

Let

\[ \text{den} = S \theta^2 M^2CL_k - \cos \gamma_k + T_w \sin \alpha_k \]

the costate equation is:

\[ \frac{\partial H_k}{\partial \alpha_k} = 0 \]

In an expanded form, this is

\[ \alpha \cdot \text{den} + \lambda_k \cdot \left( S \theta^2 M^2CL_0 \sin \gamma_k + T_w \cos \alpha_k \right) \cdot \mu_k + \mu_k \cdot \alpha_k \]

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 \( \mu \) 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 \( \alpha \) neural network to find optimal \( \alpha \) and integrate until \( y_1 \) for \( \alpha \) network is reached; use the \( M_0 \) values to find \( \alpha \) from the \( \alpha \) neural network and integrate until \( y_1 \) is reached, and so on, until \( y_1 \) 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.
5. NUMERICAL RESULTS

Tables of aerodynamic data of $C_a$ and $C_n$ 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 number 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


