15th Triennial World Congress, Barcelona, Spain

OPTIMAL MIDCOURSE GUIDANCE LAW WITH NEURAL NETWORKS

Dongchen Han*, S.N.Balakrishnan*,E.J Ohlmeyer**

Department of Mechanical and Aerospace Engineering and Engineering Mechanics
University of Missouri-Rolla, Rolla,MO 65409-0050
***Naval Surface Warfare Center, Dahlgren,Va*

Abstract: A neural-network-based synthesis of an optimal midcourse guidance law is presented in this study. We use a set of two neural networks; the first network called a "critic" outputs the Lagrange's multipliers arising in an optimal control formulation and second network, called an "action" network, outputs the optimal guidance/control. The system equations, the optimality conditions, the costate equations are used in conjunction with the network outputs to provide the targets for the neural networks. When the critic and action network are mutually consistent, the output of the action network yields optimal guidance/control. Numerical results for a number of scenarios show that the network performance is excellent. Corroboration for optimality is provided by comparisons of the numerical solutions using a shooting method for a number of scenarios. *Copyright © 2002 IFAC*

Keywords: Aerospace, Optimal Control, Neural Networks.

1. INTRODUCTION

Midcourse guidance considered in this study deals with scenarios wherein a surface launched missile seeks to intercept an airborne target. In an optimal setting, the resulting trajectory seeks to maximize the pursuer velocity at the time of intercept. Two types of guidance laws have been popular in the midcourse guidance literature (Ohlmeyer, 1994); they are the Kappa guidance and "explicit" guidance approaches. A linearized Kappa guidance (Serakos and Lin, 1996) is proposed in which a coordinate transformation is used. All of these guidance laws use some sort of approximations to the equations of motion, which is basically nonlinear. The use of neural guidance laws as used in this study, however, allow the use of the nonlinear equations directly without the need for any approximation.

In this paper an "adaptive critic" neural-network-based guidance law is proposed which: 1) solves any system (linear or nonlinear) without any approximation; 2) yields a guidance law in a feedback form as a function of current states; 3) maintains the same structure regardless of the type of system; 4) and when implemented in practice, it generates the guidance law almost instantly. Such a formulation is afforded by the "adaptive critic" architecture. The reason for choosing this structure for formulation the optimal control problem are that this approach needs no external training as in other form of neural controllers. This is not an open-loop optimal guidance law, but a feedback guidance law. Balakrishnan and Biega have shown the usefulness of this architecture for infinite-time linear problem (Balakrishnan and Viega, 1996). Han and Balakrishnan further applied this method to the agile missile control problem (Balakrishnan and Han, 1998), (Balakrishnan and Han, 1999a), (Balakrishnan and Han, 1999b). This study is very different in the sense in that this is the first time such an approach is used in the midcourse guidance literature and this is also the first guidance example with vector inputs and variable final flightpath angles.

2. PROBLEM FORMULATION AND SOLUTION DEVELOPMENT

2.1 Midcourse Course Guidance Scenario in A Vertical Plane

Derivation of Optimal Guidance Law

In seeker coordinates shown in Figure 1, a parameter κ is defined as

$$\kappa = \frac{d\gamma}{ds} = \frac{1}{V}\frac{d\gamma}{dt} \tag{1}$$

where:

γ - flightpath angle
V - velocity
s - arc length along the trajectory in Figure 3.
R - relative range between missile and predicted impact point (PIP).
θ - elevation angle of range vector measured from local horizontal.
δ - heading error

In this system, the equation of motion for the missile can be written as

$$\dot{\gamma} = V \cdot \kappa \tag{2}$$

$$\dot{R} = -V \cos\delta \tag{3}$$

$$\dot{\theta} = \frac{V \ Sin \ \delta}{R} \tag{4}$$

$$\delta = \gamma + \theta \tag{5}$$

Using the range R rather than time as the independent variable, we can reformulate as:

$$\frac{d\gamma}{dR} = -\kappa \sec \delta \tag{5a}$$

$$\frac{d\delta}{dR} = \frac{V\kappa + V\sin\delta/R}{-V\cos\delta} = -\kappa\sec\delta - \frac{\tan\delta}{R} \tag{6}$$

$$\frac{dt}{dR} = -\frac{1}{V\cos\delta} - 90 < \delta < 90^{\circ} \tag{7}$$

The main objective in a midcourse guidance is to maximize the final velocity at the predicted impact point. Hence, an appropriate cost function J is defined as

$$J = -\int_{\gamma_0}^{\gamma_f} [\frac{d(ln\ V)}{d\gamma}] d\gamma = -ln(\frac{V_f}{V_0}) \tag{8}$$

where the subscripts 0 and f denote initial and final condition respectively.
From Eqn.(15), we get:

$$V_f = V_0 e^{-J} \tag{9}$$

So maximizing V_f is equivalent to minimizing J. After some algebra the cost function can be obtained as

$$J = -\int_R^0 [\frac{\kappa^2}{2} + \omega^2] \sec \delta \cdot dR \tag{10}$$

where $$\omega^2 = \frac{k_0 k_1 \rho^2}{2}[\frac{T/m}{k_1 \rho v^2} - \frac{k_0}{k_1} + 1]^2 \tag{11}$$

$$k_0 = \frac{SC_{D0}}{2m} \tag{12}$$

$$k_1 = \frac{SC_{N\alpha}}{2m} \tag{13}$$

T is the thrust of the missile, ρ, *the* air density, S, *the* reference area of the missile, m, *the* mass, C_{D0}, zero lift drag coefficient, $C_{N\alpha}$, normal force coefficient derivative with respect to angle of attack, α. ω is a parameter representing the missile characteristics. It is common in midcourse guidance literature to treat ω as a constant.

Application of the Minimum Principle:

In our case, we use p as independent variable rather than R. p is the distance from the missile position to launch point. The relation between p and R is

$$R = R_0 - p \tag{14}$$

In order to use adaptive-critic based neural network to get the optimal solution to above problem, it is necessary to use a discrete state equation and the associated optimal control:

1. state equations:

$$\delta_{k+1} = \delta_k + (\kappa_k \sec\delta_k + \frac{\tan\delta_k}{R_0 - p_k})\Delta p_k \tag{15a}$$

$$\gamma_{k+1} = \gamma_k + \kappa_k \sec\delta_k \Delta p_k \tag{15b}$$

2. Cost function:

$$\min J = \sum_{k=0}^{N-1} (\omega_k^2 + \kappa_k^2/2)\sec\delta_k \cdot \Delta p_k \tag{16}$$

Define the Hamiltonian:

$$\begin{aligned} H_k = {} & \lambda_{1k+1}(\delta_k + (\kappa_k \sec\delta_k + \frac{\tan\delta_k}{R_0 - p_k})\Delta p_k) + \\ & \lambda_{2k+1}(\gamma_k + \kappa_k \sec\delta_k \Delta p_k) + \\ & (\omega_k^2 + \kappa_k^2/2)\sec\delta_k \cdot \Delta p_k \end{aligned} \tag{17}$$

3. The costate equations are

$$\begin{aligned} \lambda_{1k} = \frac{\partial H_k}{\partial \delta_k} = {} & \lambda_{1k+1} + \lambda_{1k+1}(\kappa_k \cdot \sec\delta_k \cdot \tan\delta_k + \frac{\sec^2\delta_k}{R_0 - p_k})\Delta p_k \\ & + \lambda_{2k+1}\kappa_k \sec\delta_k \tan\delta_k \Delta p_k + (\omega_k^2 + \frac{\kappa_k^2}{2})\sec\delta_k \tan\delta_k \Delta p_k \end{aligned}$$

$$\lambda_{2k} = \frac{\partial H_k}{\partial \gamma_k} = \lambda_{2k+1} \tag{18}$$

4. The optimality control condition is

$$\frac{\partial H_k}{\partial \kappa_k} = 0 \Rightarrow \quad \kappa_k = -\lambda_{1k+1} - \lambda_{2k+1} \tag{19}$$

5. The boundary conditions are: $p_0 = 0, \quad \delta_0, \gamma_0$ are known

$p_N = R_0, \quad \delta_N = 0, \gamma_N$ are fixed values. Note that $Đp$ is the stepsize and k denotes the stage.

3. DEVELOPMENT OF A NEURAL NETWORK SOLUTION

In this paper, neurocontrollers are obtained for both fixed final flightpath angle and flexible flightpath angle. After discretizing the system equation, the independent variable p will be divided into appropriate steps. During each period, two neural networks, namely the "action" network which represents state feedback guidance law and output control κ, and another network called "critic" network which represents the supervisory model will output costate λ_1, λ_2. The trained action networks which are cascaded together will form the optimal guidance law when implemented in-real time.

3.1 Procedure to Train Neural Networks

For "finite time" (or finite-horizon) problems, the "time" is fixed. In our case the independent variable p is a value which is determined by tactical requirements. Here, according to Navy's specification, p_N is chosen as 60 miles. Assume p is divided into *N-1* fragment periods. Then, there are totally N steps. The networks are synthesized backwards in this formulation. This procedure includes two stages:

Synthesis of the Last Network:

1. Randomly pick δ_{N-1}, since $\delta_N = 0$, given Δp_{N-1}, from state Eqn.(15a), κ^*_{N-1} is obtained.
2. Since γ_N is fixed, so input δ_{N-1} and κ_{N-1} into Eqn.(15b), γ_{N-1} can be obtained.
3. Train a network denoted as $\kappa(N-1)$:inputs are state δ_{N-1} and γ_{N-1}, target is κ^*_{N-1}. Train this network until error performance is satisfied.
4. Pick λ_{1N} or λ_{2N}, together with κ_{N-1}, input into Eqn.(18) to obtain λ_{2N} or λ_{1N}.
5. Input $\lambda_{1N}, \lambda_{2N}, \kappa_{N-1}, \Delta p_{N-1}, \delta_{N-1}, \omega_{N-1}$ into Eqns.(19) to obtain $\lambda^*_{1N-1}, \lambda^*_{2N-1}$.
6. Train a network called $\lambda(N-1)$: inputs are state δ_{N-1} and γ_{N-1}, targets are $\lambda^*_{1N-1}, \lambda^*_{2N-1}$. Train this network until a specified error performance is satisfied.

Synthesis of Other Networks:

7. Determine Δp_{N-2}, pick $\delta_{N-2}, \gamma_{N-2}$, input into $\kappa(N-1)$ network to obtain κ_{N-2}.
8. Input $\delta_{N-2}, \gamma_{N-2}, \kappa_{N-2}$ and Δp_{N-2} into Eqns.(15a), (15b) to obtain $\gamma_{N-1}, \delta_{N-1}$.
9. Input $\gamma_{N-1}, \delta_{N-1}$ into $\lambda(N-1)$ network to obtain $\lambda_{1N-1}, \lambda_{2N-1}$.
10. Input $\lambda_{1N-1}, \lambda_{2N-1}$ into Eqn.(19) to obtain κ^*_{N-2}.
11. Train a $\kappa(N-2)$ network with inputs $\delta_{N-2}, \gamma_{N-2}$ and targets κ^*_{N-2}, until convergence is reached.
12. Input $\delta_{N-2}, \gamma_{N-2}$ into $\kappa(N-2)$ network to obtain κ_{N-2}.
13. Input $\delta_{N-2}, \gamma_{N-2}$ and κ_{N-2} into Eqns.(15a), (15b) to obtain $\gamma_{N-1}, \delta_{N-1}$.
14. Input $\gamma_{N-1}, \delta_{N-1}$ into $\lambda(N-1)$ network to obtain $\lambda_{1N-1}, \lambda_{2N-1}$.
15. Input $\lambda_{1N-1}, \lambda_{2N-1}, \kappa_{N-2}, \delta_{N-2}, \gamma_{N-2}$ into Eqns.(19) to obtain target $\lambda^*_{1N-2}, \lambda^*_{2N-2}$.
16. Train a $\lambda(N-2)$ network with inputs $\delta_{N-2}, \gamma_{N-2}$ and targets λ^*_{1N-2}, λ^*_{2N-2}, until convergence is reached.
17. Repeat the process 7 – 16 with *N=N-1* until p_0=0 is reached. A schematic of the network development is presented in Figure 2.

4. USE OF NETWORKS AS CONTROL LAW IN REAL-TIME

Assume any δ_0, γ_0 (within the trained scope), use $\kappa(0)$ network to find κ_0 and integrate to get δ_1, γ_1 until p_1 is reached. Then input δ_1, γ_1 into $\kappa(1)$ network to get κ_1 and integrate to get δ_2, γ_2 until p_2 is reached. Continue until $p_N = R_0$ is reached.

5. SOLUTION WITH A SHOOTING METHOD

In order to verify the optimality of the solutions obtained with neural networks, The two-point boundary value problem (TPBVP) resulted from optimal control formulation are also solved with a shooting method (Bryson and Ho, 1975).

6. NUMERICAL RESULTS

6.1 Fixed Final Flightpath Angle

Results from simulations using the neural network approach and the shooting method are presented in this section. The desired final states for the midcourse missile were fixed at zero for the heading error δ, zero for the flightpath angle γ, and sixty miles for the range. The parameter ω was set at 4.0E-5. A feedforward neural network with three layers with a hyperbolic tangent sigmoid, a log-sigmoid, and a linear activation function is used for the controller network as well as for the critic network. The number of neurons is 4 in the first and second layer and 1 in third layer. The Levenberg-Marquardt Backpropagation algorithm is used in training both the action and the critic neural networks. These choices for structure and training method are intuitive and are not necessary optimal. In the training process, a variable stepsize Runge-Kutta numerical method is used to integrate the state equations between the steps defined by the intervals in the independent variable. After training, 62 pairs of networks are obtained. It should be noted that this means that the optimal guidance law has been obtained with starting from **any range** from 0 to 60 miles to reach the predicted impact point with maximum velocity and with zero heading error and zero flightpath angle.

Three-dimensional plots of trajectories obtained with the neural networks are presented in Fig. 3. The two costates (Lagrange's multipliers) are plotted in Fig. 4. The corresponding history of the control variable κ is presented in Fig. 5. In order to verify the optimality of the neural network results, the same initial conditions were used and the shooting method was used 36 times by solving each single problem separately. Although they are not superposed with the neural network results (because we want to show optimality of the neural network results over the entire range and the superposition of plots will make them very busy), we observed that the state vector histories, the costate histories, and the control histories are almost identical to the corresponding variable histories obtained from neural networks. Note that the neural networks embed countable infinite optimal solutions to the midcourse guidance problem with an envelope of initial conditions as can be observed from the initial conditions in Fig. 3 or with any range up to 60 miles.

6.2. Varied Final Flightpath Angle

The results presented in previous section were obtained for a fixed final flightpath angle $\gamma_f = 0$. In order to capture the target in a different situation, it is necessary for the missile to reach the PIP at different final flighpath angles or from different attitudes. If we use one set of networks for each final flightpath angle, there will be several similar networks corresponding to different final flightpath angles. For any other final flightpath angle, one has to interpolate two successive neural network controllers. This is not practical in real implementation since that will result in a higher cost and will be less accurate. So we exploit the neural network's universal learning or mapping ability. One single network will be used to output costates for different final flightpath angles at any stage. The way to do this is to augment final flightpath path angle as another input and target during neural network training. Here the final flightpath angle is chosen from 60^0 to 90^0 with the interval 5^0 as the training scope. The initial heading error and flightpath angle scope remain the same as above. The range and the parameter ω were assumed to be same as before. In this new design a feedforward neural network was chosen with all three layers with linear activation functions for the controller network as well as for the critic network. The numbers of neurons are 8 in the first layer and 8 in the second layer for both the controller and critic networks. We still used a Levenberg-Marquardt algorithm in training both the action and the critic neural networks. This time 79 pairs of networks were used. The results of neural network controller for a set of initial conditions and desired final flightpath angles are plotted in Fig. 6-8. In every case the neurocontroller takes the missile to the desired final flightpath angle. In comparing the neurosolutions with the shooting method, the maximum error for δ is about 3.5^0 during whole procedure and the error at the end approaches zero. The maximum error for γ is about 4.3^0 and the error approaches zero at the end. The maximum κ error is about 0.016 at the end. The errors can be attributed to the different step-size. The larger error trajectories are those which have smaller initial flightpath angle and heading angle but larger negative final flightpath angle. The reason is that it is more difficult to shape a low, flat trajectory at the final stage.

7. CONCLUSIONS

A neural network approach to solving optimal midcourse guidance problems has been presented in this study. This approach solves nonlinear guidance/control problems without making approximations to the model. The results show that the adaptive critic-based neural networks present a powerful computational approach to such class of problems.

8. ACKNOWLEDGMENT

This research was supported by Draper Laboratories grant DL-H-513245.

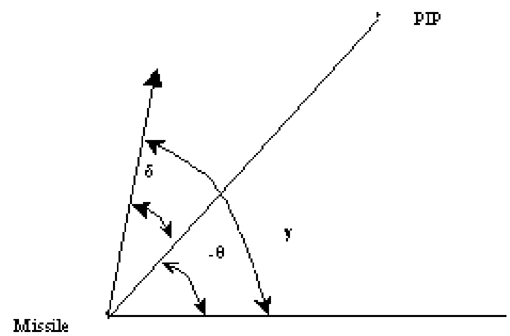

Fig. 1. Missile Intercept Scenario

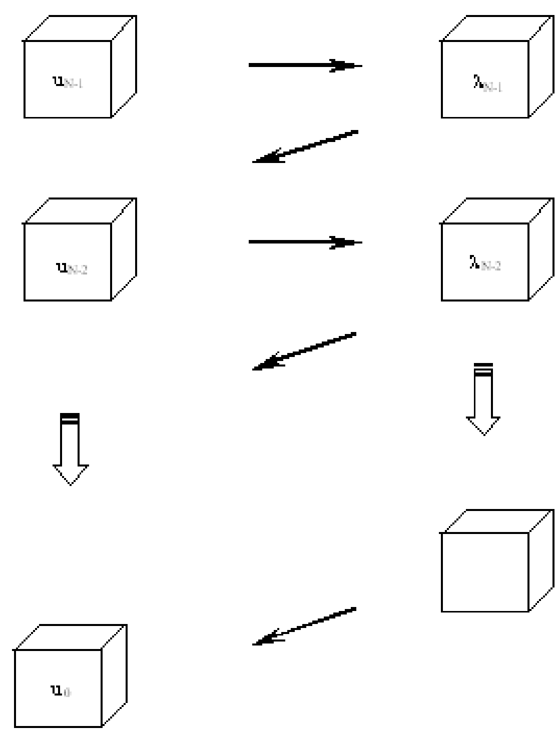

Fig. 2. Schematic of Successive Adaptive Critic Synthesis

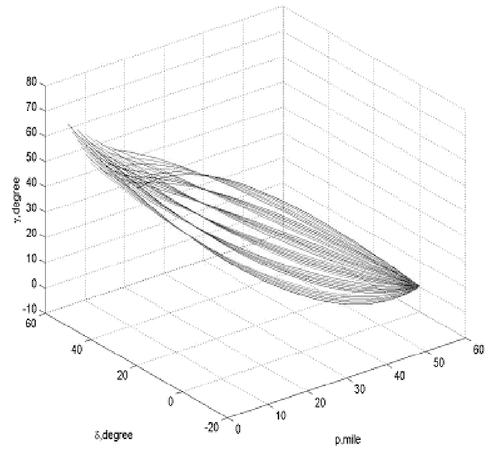

Fig. 3. Trajectories of δ, γ (NN)

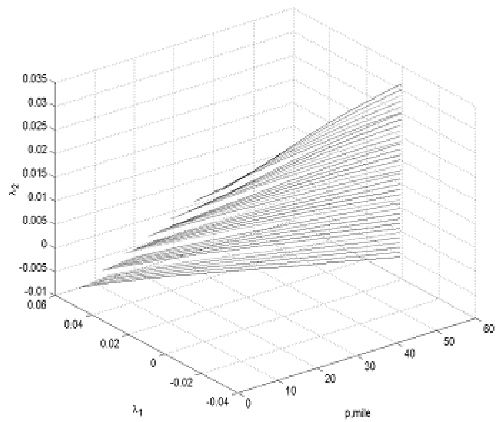

Fig. 4. Costate λ Trajectories

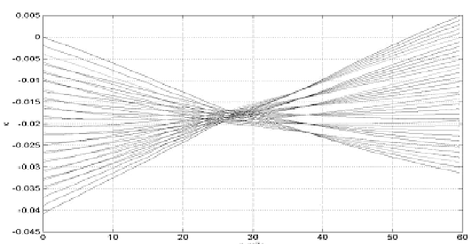

Fig. 5. Kappa Trajectories (NN)

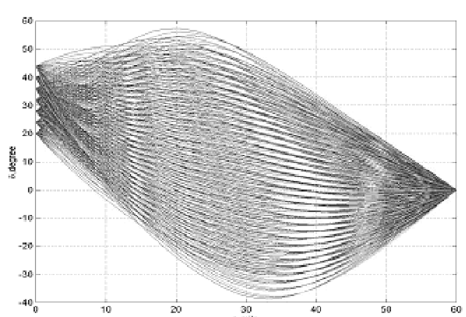

Fig. 6. δ Trajectories for Different γ_f (NN)

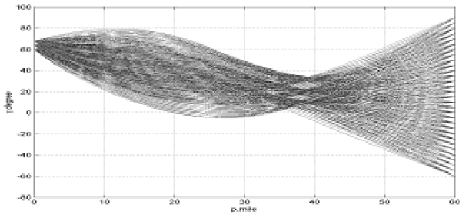

Fig. 7. γ Trajectories for Different γ_f (NN)

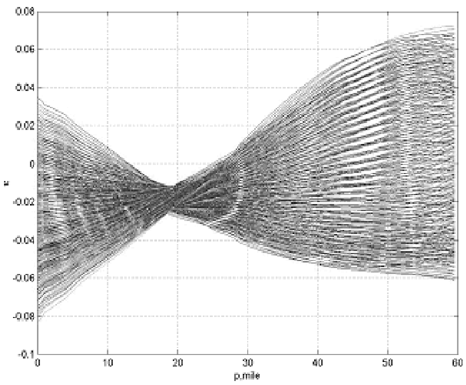

Fig. 8. Kappa Trajectories for Different γ_f (NN)

REFERENCES

Balakrishnan S.N. and V.Biega (1996). Adaptive Critic Based Neural Networks for Aircraft Optimal Control. *Journal of Guidance, Control, and Dynamics*. **Vol. 19, No .4**, July-Aug, 1999, pp893-898.

Balakrishnan, S.N. and D. Han (1998). *Robust Adaptive Critic Based Neural Network for Agile Missile Control*. AIAA Guidance, Navigation, and Control Conference and Exhibit, Boston, Massachusetts, August 10 – 12,1998,pp1803-1812.

Balakrishnan, S.N. and D.Han (1999a). *Robust Adaptive Critic Based Neural Network for Control-constrained Agile Missile Control*. American Control Conference. June 1999, San Diego, California. pp2600-2604.

Balakrishnan, S.N. and D. Han (1999b). *Robust Adaptive Critic Based Neural Network for Speed-constrained Agile Missile Control*. AIAA Guidance, Navigation, and Control Conference and Exhibit, August 9 – 11,1999, Portland, Oregon.

Bryson, A.E and Y.C.Ho (1975) *Applied Optimal Control*, Halsted Press.

Ohlmeyer, E.J. (1994). *Standard Missile Midcourse Guidance Laws*, NSWCDD internal report, Naval Surface Warefare Center, Dahlgren Division, Dahlgren, VA 22448,April 1994.

Serakos, D. and C.F.Lin (1996). *Linearized Kappa Guidance*, NSWCDD report, Naval SurfaceWarfare Center, Dahlgren Division, Dahlgren, VA 22448-5100,Aug 1996