01 Jan 1995

A New Neural Architecture for Homing Missile Guidance

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

Victor Biega

Recommended Citation

S. N. Balakrishnan and V. Biega, "A New Neural Architecture for Homing Missile Guidance," *Proceedings of the 1995 American Control Conference*, Institute of Electrical and Electronics Engineers (IEEE), Jan 1995.
The definitive version is available at https://doi.org/10.1109/ACC.1995.531278

TM7 - 1:35

Proceedings of the
American Control Conference
Seattle, Washington • June 1995

A NEW NEURAL ARCHITECTURE FOR HOMING MISSILE GUIDANCE

S. N. Balakrishnan[1] and Victor Biega[2]
Department of Mechanical and Aerospace Engineering
and Engineering Mechanics
University of Missouri-Rolla
Rolla, MO 65401-0249
bala@umr.edu

I. Introduction

Improved guidance or trajectory design can lead to increased missile performance by flying a more optimal trajectory. The increased missile performance is characterized in terms of its mission. It could be the achievement of maximum velocity or maximum energy if it is the midcourse guidance of a tactical missile; it can be measured in terms of the terminal miss distance for an air to surface, air to air, or surface to air missiles.

Most of the short range missiles use pronav to guide them in the terminal phase. Even though the pronav performs well in several scenarios, its performance degrades if the target is highly maneuverable or if the boresight angle is large. Several modifications have been suggested in the literature for improvement/compensation. These include a constant bias to the measured line-of-sight before calculation of the commanded acceleration. For a detailed list of references on guidance and control of air-to-air missiles, refer to Cloutier at al. [1989].

An approximately optimal guidance law which minimizes the kinetic energy loss is proposed by Glasson and Mealy [1983]. Their strategy uses a time-scheduled navigation ratio. Cheng and Gupta [1986] use singular perturbation theory to develop a guidance law which minimizes flight time. In the study by Menon and Briggs [1987], the cost function minimizes flight time and the specific energy at the final time. The singular perturbation approach in these studies are based on defining slow dynamics (cross range, flight path angle and specific energy), medium dynamics (altitude), and fast dynamics (pitch angle and yaw angle). Katzir et al. [1989] formulated near-optimal guidance to be used for real-time calculations. It is based on a neighboring optimal control concept wherein a complementary control is calculated to be added to the precalculated nominal optimal control.

Optimization is a primary concern in all these studies. Two-point boundary value problem (TPBVP) methods provide exact solutions but must be solved for each set of initial conditions. This requires determining a separate solution for each possible initial condition for a given system. Dynamic programming is also an exact method of determining optimal control for a family of conditions. This method of solution, however, becomes very difficult to solve for in higher dimension and nonlinear systems. Other methods of solution also have their advantages and disadvantages. Neighboring optimal control is beneficial in that the solution of a single TPBVP allows an approximate solution over a range of initial conditions. The disadvantage is that approximation methods such as neighboring optimal control can fail at a distance from the original TPBVP solution. In this study, we present a neural architecture to solve a typical optimal homing missile guidance problem.

There is a multitude of neurocontrollers in the published literature [White and Sofge, 1992]. Almost all of them fall within four categories: 1) supervised control, 2) direct inverse control, 3) neural adaptive control, and 4) backpropagation through time. A fifth and rarely studied class of controller has the most interesting structure. It is called an Adaptive Critic Architecture. We chose this structure for formulating the optimal control problems. The reasons are: 1) this structure obtains an optimal controller through solving dynamic programming equations, and 2) this approach has a supervisor (critic) which critiques the outputs of the controller network and a controller. Therefore, this approach has a built-in

1 Professor
2 Graduate Student

fault tolerance, 3) this approach needs NO external training as in other forms of neurocontrollers, and 4) this is not an open loop optimal controller but a feedback controller.

The method discussed in this study determines an optimal control law for a system by successively adapting two networks - an action and a critic network. This method determines the control law for an entire range of initial conditions. In addition the control law does not need to be determined mathematically. This method simultaneously determines and adapts the neural networks to the optimal control policy for both linear and nonlinear systems. In addition, it is important to know that the form of control does not need to be known in order to use this method.

II. Problem Formulation

A. Statement of the General Problem

In this study, a problem of the form

$$J=\phi(x(t_f))+\int_0^{t_f}\psi(x(\tau),u(\tau))\,d\tau \qquad (1)$$

$$\dot{x}=f(x,u) \qquad (2)$$

$$t_f\equiv given \qquad x_o\equiv given \qquad (3)$$

is being considered.

The method used in this study has advantages over the previous methods in that solutions are found over any user specified range of x, and these solutions are then available for the entire span of x. In addition, the user need not assume any predetermined form or function for the control law.

B. Dynamic Programming Background (Exact Results)

We can rewrite Eq. 1 as

$$J(x(t))=U(x(t),u(x(t)))+\langle J(x(t+1))\rangle \qquad (4)$$

Here, J(x(t)) is the cost associated with going from time t to the final time. U(x(t),u(x(t))) is the utility, which is the cost from going from time t to time t+1. Finally, $\langle J(x(t+1))\rangle$ is assumed to be the minimum cost associated with going from time t+1 to the final time.

If both sides of the equation are differentiated and we define

$$\lambda(x(t))\equiv\frac{\delta J(x(t))}{\delta x(t)} \qquad (5)$$

then

$$\lambda(x(t))=\frac{\delta U(x(t),u(t))}{\delta x(t)}+\frac{\delta U(x(t),u(t))}{\delta u(t)}+\left\langle\lambda(x(t+1))\frac{\delta x(t+1)}{\delta x(t)}\right\rangle+\left\langle\lambda(x(t+1))\frac{\delta x(t+1)}{\delta u(t)}\frac{\delta u(x(t))}{\delta x(t)}\right\rangle \qquad (6)$$

From this it can be seen that if $\langle\lambda(x(t+1))\rangle$, U(x(t),u(t)) and the system model derivatives are known then $\lambda(x(t))$ can be found.

Next, the optimality equation is defined as

$$\frac{\delta J(x(t))}{\delta u(t)}=0 \qquad (7)$$

Dynamic programming uses these equation to aid in solving an infinite horizon policy or to determine the control policy for a finite horizon problem.

C. Training Methods (Approximation Techniques)

As mentioned earlier, this study uses Eq. 7 in order to determine the optimal control policy. The basic training takes place in two stages, the training of the action network (the network modeling u(x(t)) and the training of the critic network (the network modeling, or approximating $\lambda(x(t))$. Both networks are assumed to be feedforward multiple layer perceptron networks.

To train the action network for time step t, first x(t) is randomized and the action network outputs u(t). The system model is then used to find x(t+1) and $(\delta x(t+1))/(\delta u(t))$. Next, the critic from t+1 is used to find $\lambda(x(t+1))$. This information is used to update the action network. This process is continued until a predetermined level of convergence is reached.

To train the critic network for the time step t, x(t) is randomized and the output of the critic $\lambda(x(t))$ is found. The action network from step t calculates u(t) and $(\delta u(t))/(\delta x(t))$. The model is then used to find $(\delta x(t+1))/(\delta x(t))$, $(\delta x(t+1))/(\delta u(t))$ and x(t+1). The critic from step t+1 is then used to find $\lambda(x(t+1))$. After this, Eq. 6 is used to find $\lambda^*(x(t))$, the target value for the critic. This process is continued until a predetermined level of convergence is reached.

D. Homing Missile Guidance

The equations of motion of a target-intercept model (Figure 1) and an appropriate cost function are presented in this section. The equations of relative motion between a constant-velocity target and a missile in two dimensions are:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \ddot{x} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_x \\ u_y \end{bmatrix} \tag{8}$$

The cost function which seeks to minimize the terminal miss-distance and the control effort is given by

$$J = \frac{1}{2} \left(x(t_f)^2 + y(t_f)^2\right) + \frac{\gamma}{2} \int_{t_o}^{t_f} u^T u \, dt \tag{9}$$

In these equations, x,y are the relative positions and $\dot{x}$ and $\dot{y}$ are the relative velocities; u_x and u_y are the missile controls in x and y directions, respectively.

The first step of this problem is to discretize the system. A time step of 0.4 seconds was chosen. Using Euler's method, this results in the following discretized system

$$\begin{bmatrix} x(t+1) \\ y(t+1) \\ \dot{x}(t+1) \\ \dot{y}(t+1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0.4 & 0 \\ 0 & 1 & 0 & 0.4 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \\ \dot{x}(t) \\ \dot{y}(t) \end{bmatrix} + \begin{bmatrix} 0.08 & 0 \\ 0 & 0.08 \\ 0.4 & 0 \\ 0 & 0.4 \end{bmatrix} \begin{bmatrix} u_x(t) \\ u_y(t) \end{bmatrix} \tag{10}$$

with the discretized cost function

$$J = \frac{1}{2} \left(x(t_f)^2 + y(t_f)^2\right) + 0.4 \frac{\gamma}{2} \sum_{t=0}^{t_f} u^T u \tag{11}$$

Next, a neural network is designed and the initial weights are randomized. After randomization, the appropriate utility functions are specified. $\lambda(t+1)$ is set equal to zero. With these definitions it is then possible to determine the control law for the final time step using a gradient descent algorithm. Next, a neural network is randomized to function as the critic for the final time step. Using the same definitions for $\lambda(t+1)$ and the utility function as the final control law allows the use of Eq. 6 to determine the final critic mapping. After the critic for the final time step has converged, the action network for the previous time step can be determined. (Note that $\lambda(t+1)$ is determined from the critic network for the final time step. This, with the utility function, allows the action network to be trained for the optimal control law for the previous time step.)

After the action network has converged, the next to the last step critic network can be determined using this new action network and the previous critic network. This information, along with Eq. 6, provides the information to determine the new critic network. This backward sweep methodology is continued to determine the action and critic networks for each time step. This process continues until the control has been determined for the desired interval of time.

III. Discussion of Numerical Results

The homing missile problem was solved for a gamma of 10^{-4}. The desired final time was assumed to be 5.2 seconds. Figure (2) shows values for x and y for both the neural network determined trajectories and the optimal trajectories determined by LQR methodology. Note that both trajectories are nearly identical. Figure (3) shows the same trajectories for the velocity in the x and y directions. Once again, notice that these trajectories are nearly identical. It is important to note that the initial conditions for this problem can be generated randomly. To show the feedback capabilities of this technique, we generated random initial conditions for the positions and velocities and ran the trajectories with control provided by the neural networks. The states, as well as the control, are presented in Figures (4-9). It can be observed that in all these cases, the resulting trajectories and control are almost identical with pointwise solutions of LQR results. Another feature of this technology is to provide optimal control at any given time-to-go. (Here it is provided by stage N). From Figures (10) and (11), we can observe that at any given time-to-go and same initial conditions, the neural networks provide control for nearly optimal trajectories. This method determined the control law not only for a specific starting point, but also for any point within the training range.

IV. Conclusions

We have presented a new neural architecture which imbeds dynamic programming solutions to solve optimal target-intercept problems. They provide feedback guidance solutions, which are optimal with any initial conditions and time-to-go, for a two dimensional scenario.

Acknowledgement

This study was funded by NSF Grant ECS-9313946 and by the Missouri Department of Economic Development Center for Advanced Technology Program. The authors would like to thank Dr. Paul Werbos for his technical comments on the neural network approach.

References

[1] Cloutier, J. R., Evers, J. H. and Feely, J. J., "Assessment of Air-to-Air Missile Guidance and Control Technology," IEEE Control Systems Magazine, October 1989, pp. 27-34.

[2] Glasson, D. P. and Mealy, G. L., "Optimal Guidance for Beyond Visual Range Missiles Vol. I," AFATL-TR-83-89, November 1983.

[3] Cheng, V. H. L. and Gupta, N. K., "Advance Midcourse Guidance for Air-to-Air Missile," Journal of Guidance and Control, Vol. 9, No. 2, March-April 1985, pp. 135-142.

[4] Menon, P. K. and Briggs, M. M., "A Midcourse Guidance Law for Air-to-Air Missiles," AIAA-81-25-09, 1987 AIAA Guidance, Navigation and Control Conference, August 1987.

[5] Katzir, S., Clift, E. M. and Lutze, F. H., "An Approach to Near Optimal Guidance On-board Calculations," IEEE ICCON, April 1989.

[6] White, D. A. and Sofge, D., "Handbook of Intelligent Control," Van Nostrand Reinhold, 1992.

Problem Definition

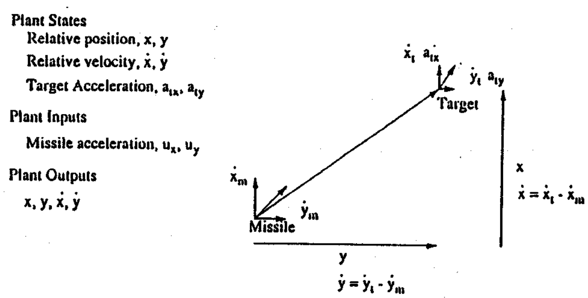

Figure 1:

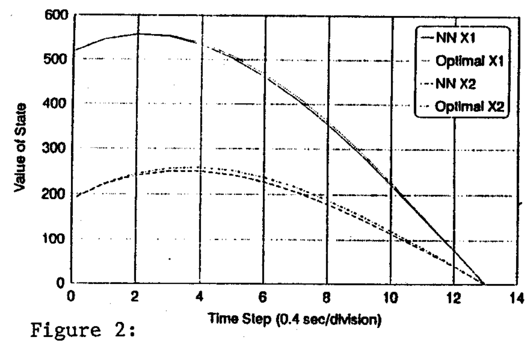

Figure 2:
Optimal and NN determined values for States x1 and x2 for the Missile Guidance Problem

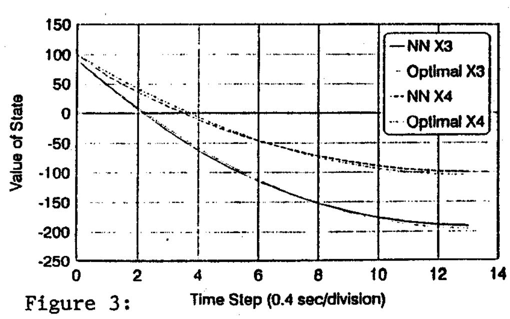

Figure 3:
Optimal and NN Determined Values for States x3 and x4 for the Missile Guidance Problem

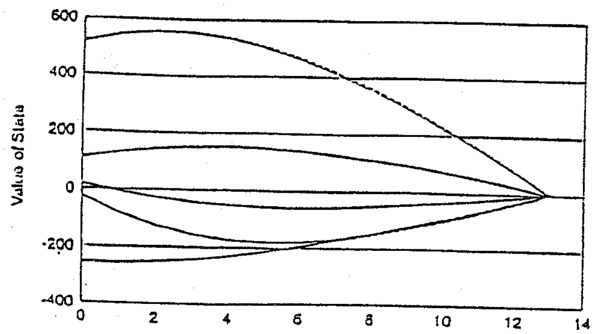

Figure 4:
Value of State x1 for Various Initial Conditions
solid- Neural Network Determined
dashed- Optimal

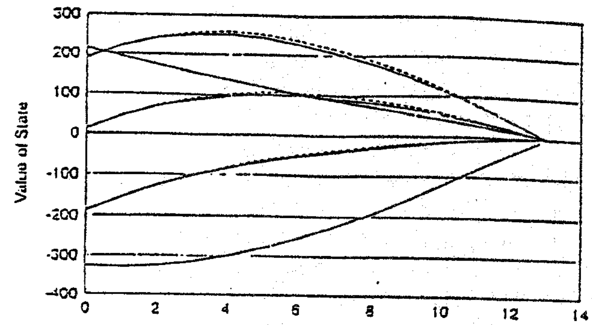

Figure 5:
Value of State x2 for Various Initial Conditions
solid- Neural Network Determined
dashed- Optimal

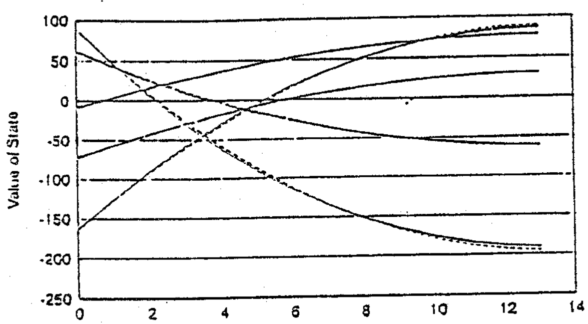

Figure 6:
Value of State x3 for Various Initial Conditions

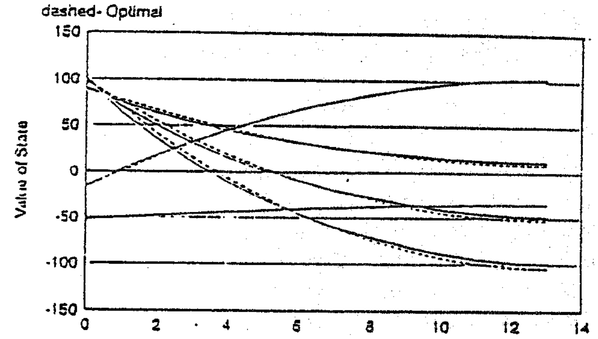

Figure 7:
Value of State x4 for Various Initial Conditions

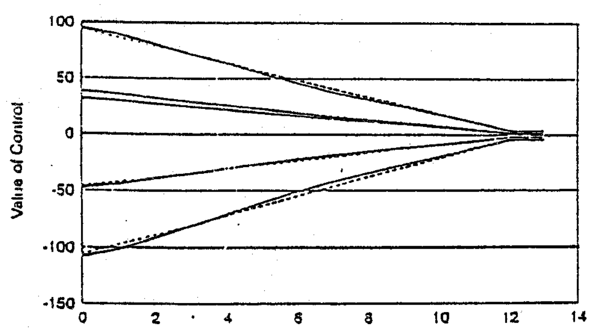

Figure 8:
Value of Control u1 for Various Initial Conditions

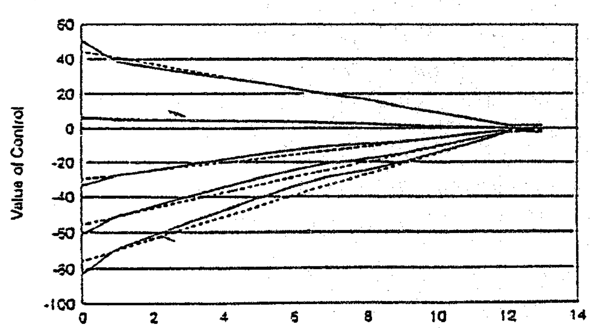

Figure 9:
Valuemof Control u2 for Various Initial conditions

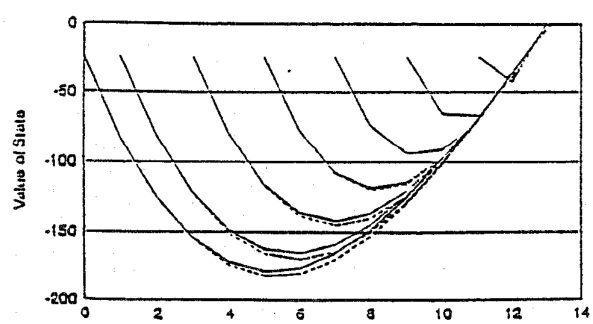

Figure 10:
Value of State for x1 for varying remaining Time Steps (I.c. x=[-24 -187 -164 90]).

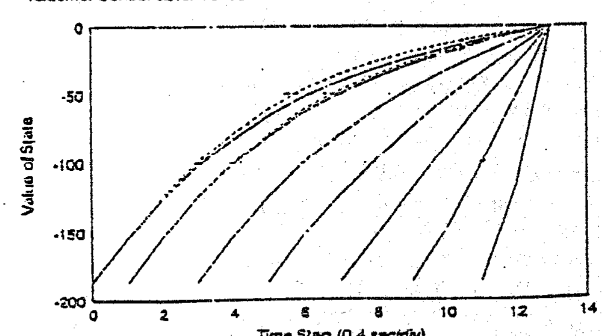

Figure 11:
Value of State for x2 for varying remaining Time Steps (I.c. x=[-24 -187 -164 90])