GAME THEORY SOLUTIONS USING NEURAL NETWORKS FOR MISSILE GUIDANCE

Venkat Durbha* and S.N. Balakrishnan**
vpdkc7@umr.edu, bala@umr.edu
Department of Mechanical and Aerospace Engineering
University of Missouri-Rolla, Rolla, MO 65401

1. INTRODUCTION

Elementary game theory is concerned with optimization of certain pay off between two contesting parties. Pursuit-Evasion games are a class of problems with conflicting interests of players. In such problems, the pay-off can be the time of intercept or the final miss distance. While the Pursuer tries to minimize the afore-mentioned pay-off, the evader tries to maximize it. This is analogous to situation involving an attacking Missile/Airplane and an Airplane/Missile trying to avoid interception. This can be considered as optimal control problem with the pay-off as the cost function. Neural networks have been successfully used to solve various optimal control problems in recent years. In the present study, Adaptive-Critic based neural networks are employed to solve a linear pursuit-evasion problem. This is achieved by successively adapting three networks, an action network each for the pursuer and the evader and a critic network common for both the pursuer and evader. The final miss distance is taken as the value to be optimized. Problems with nonlinear dynamics are in progress.

This paper is divided into five sections. In Section 2, there is a brief description of the mathematical model for a linear pursuit-evasion game. This is based on basic two person game described in Bryson[1]. The third section gives an overview of the applications of Adaptive Critic based neural networks for solving both linear and nonlinear optimal control problems. The fourth section explains the steps involved in adapting the neural network for a finite time problem such as this. Section five discusses the results obtained and the further work being done.

* Graduate student
** Professor, Associate fellow, AIAA, Contact Author
2. LINEAR PURSUIT-EVASION GAME

Consider a scalar linear pursuit-evasion game described in [1]. The dynamics of Pursuer and Evader are given by

\begin{align}
\dot{x}_p &= A_p x_p + b_p u \\
\dot{x}_e &= A_e x_e + b_e v 
\end{align}

(1)

where \( p \) represents the pursuer and \( e \) represents the evader. \( x_p \) is the position of the pursuer and \( x_e \) is the position of the evader. \( u \) and \( v \) are the control available to the pursuer and the evader respectively. \( A_p \) and \( b_p \) relate the velocity of the pursuer with its position and control respectively. \( A_e \) and \( b_e \) relate the velocity of the evader with its position and control. The final miss by the pursuer is taken as the cost to be minimized. This is given by the weighed quadratic form Eq (2)

\[ \left\| x_p(t_f) - x_e(t_f) \right\|_{M^TM}^2 \]

(2)

where \( M^TM \) is the weight applied to terminal distance between pursuer and evader.

It is to be noted that the objective of the evader is to maximize this cost function. In real life situations, the pursuer is required to come as close to the evader within a fixed final time. The control variables of both the pursuer and the evader are limited by the integral quadratic constraint (3),

\begin{align}
\int_{t_0}^{t_f} \left\| u \right\|_{R_p}^2 dt &\leq E_p \\
\int_{t_0}^{t_f} \left\| v \right\|_{R_e}^2 dt &\leq E_e 
\end{align}

(3)

where \( E_p \) and \( E_e \) are the maximum control available to pursuer and evader respectively. \( R_p \) is the weight on the pursuer control and \( R_e \) is the weight on evader control.
Both the pursuer and the evader would use maximum control available to achieve their desired objective. However, it is assumed, that the control available to the pursuer is greater than that available to the evader. This condition is essential for any degree of success in interception. The constraints on the available control are added to the cost function. The augmented cost function with the above constraints is given by Eq (4),

\[
J = \frac{1}{2} \left\| x_p(t_f) - x_e(t_f) \right\|_{M't'M}^2 + \frac{1}{2} \int_{t_o}^{t_f} \left( \left\| u \right\|_{R_p}^2 - \left\| v \right\|_{R_e}^2 \right) dt
\]  

(4)

As the evader tries to maximize the J value its constraint is subtracted. For the purpose of analysis the distance between the pursuer and evader is made as state variable denoted by \( z(t) \) as given in Eq (5).

\[
\begin{align*}
&z(t) \triangleq M (\hat{x}_p(t) - \hat{x}_e(t)) \\
&\hat{x}_p(t) \triangleq \Phi_p(t_f,t)x_p(t) \\
&\hat{x}_e(t) \triangleq \Phi_e(t_f,t)x_e(t)
\end{align*}
\]

(5)

where \( \Phi_p(t_f,t) \) represents the fundamental matrix of \( A_p \) and \( \Phi_e(t_f,t) \) represents the fundamental matrix of \( A_e \).

Therefore the new state equation and the cost function can be simplified to,

\[
\dot{z}(t) = \rho(t)u - \epsilon(t)v
\]

(6)

\[
J = \frac{1}{2} \left\| z(t_f) \right\|^2 + \frac{1}{2} \int_{t_o}^{t_f} \left( \left\| u \right\|_{R_p}^2 - \left\| v \right\|_{R_e}^2 \right) dt
\]

(7)

where,

\[
\begin{align*}
&\rho(t) = M \Phi(t_f,t)b_p \\
&\epsilon(t) = M \Phi(t_f,t)b_e
\end{align*}
\]

(8) (9)

The Hamiltonian for the modified state equation is given by

\[
H = \frac{1}{2} (u^TR_pu - v^TR_ev) + \lambda^T (\rho(t)u - \epsilon(t)v)
\]

(10)

where, \( \lambda \) is the Lagrange multiplier.
The co-state equations and the optimality conditions are as follows

\[ \dot{\lambda} = 0 ; \quad \lambda(t_f) = z(t_f) \]  

(11)

\[ u = -R_r^{-1}\rho^T\lambda ; \quad v = -R_r^{-1}\epsilon^T\lambda \]  

(12)

3. NEURAL NETWORKS FOR CONTROL

It is well known that the dynamic programming formulation offers the most comprehensive solution to nonlinear optimal control; however, a huge amount of computational and storage requirements are needed to solve the associated Hamilton-Jacobi-Bellman (HJB) equation [Bryson and Ho, 1975] (also known as the Bellman equation). Werbos [1992] proposed a means to get around this numerical complexity by using 'approximate dynamic programming' (ADP) formulations. His methods approximate the original problem with a discrete formulation. The solution to the ADP formulation is obtained through the two-neural network adaptive critic approach. In one version of the adaptive critic approach called the dual heuristic programming (DHP) one network called the action network represents the mapping between the state variables of a dynamic system and control and the second network, called the critic, outputs the costates with the state variables as its inputs. This ADP process, through the nonlinear function approximation capabilities of neural networks, overcomes the computational complexity that plagued the dynamic programming formulation of optimal control problems. More important, this solution can be implemented on-line, since the control computation requires a few multiplications of the network weights which are trained off-line. This technique was applied by Balakrishnan and Biega[1996] to various linear and non-linear problems including an aircraft control problem. Note that there are various types of adaptive critic designs available in literature. An interested reader can refer to [Prokhorov and Wunsch, 1997] for more details on ADP and DHP.

Several authors have used neural networks to "optimally" solve nonlinear control problems [4,5,6]. For example, Kim and Calise[7] have proposed a neural network based control correction based on Lyapunov theory. A major difference between their approach and this study is that the development of guidance law/control is based on optimal control; hence, it is stabilizing and at the same time minimizing a cost.
In this study, a cascade of dual neural networks is used as a framework for the solutions of linear as well as nonlinear, finite-time optimal control problems with a special application to the pursuer-evader problems using differential game theory. In a typical adaptive critic design, the controller output does not depend the current time but only the current states; in this problem, by contrast, the controller output has to be different for the same values of the state since the time left to complete the task plays a role in how much control has to be used. Hence, a cascade of controllers is synthesized by indexing the independent variable. Han and Balakrishnan [1999,2002] further applied this method to an agile missile control problem. The contribution of this work is that we extend this idea to solve problems cast in a differential games setting where no neural network approach has been used before. In the current problem of interest, two action networks have been used at each stage. One action network is trained to generate the control for the pursuer and the other, for the evader. There is a single critic network to generate the costate at each stage. The relative distance between the pursuer and evader is the input for both the action and the critic networks. The problem is formulated as a finite time problem. Therefore the training proceeds backwards.

4. DEVELOPMENT OF NEURAL NETWORK SOLUTIONS

Development of the neural network solution is outlined in this section. The finite time is split into N steps. Note that the solution represents a backward sweep process.

**Last Network:**
1. For random values of final miss distance $Z_N$, calculate $\lambda_N$ from the final costate condition.
2. Use optimality condition $H_u=0; \ H_v=0$, Eq.(12), to solve for appropriate $U_{N-1}$ and $V_{N-1}$.
3. From the costate propagation equation, calculate $\lambda_{N-1}$.
4. Train three neural networks: The $U_{N-1}$ network outputs $U_{N-1}$, and the $V_{N-1}$ network outputs $V_{N-1}$ for different values of $Z_{N-1}$. The $\lambda_{N-1}$ network outputs $\lambda_{N-1}$ for different values of $Z_{N-1}$. We have optimal $U_{N-1}$, $V_{N-1}$ and $\lambda_{N-1}$ now.

**4.2 Other Networks:**
5. Assume different values of $Z_{N-2}$ and use random neural networks (or initialized with $U_{N-1}$ network and $V_{N-1}$ network) called $U_{N-2}$, $V_{N-2}$ networks to output $U_{N-2}$ and $V_{N-2}$. Use
$Z_{N-2}, U_{N-2}$ and $V_{N-2}$ to obtain $Z_{N-1}$. Input $Z_{N-1}$ to $\lambda_{N-1}$ network to get $\lambda_{N-1}$. Use $Z_{N-2}, \lambda_{N-1}$ in $H_u=0; H_v=0$ to solve for $U_{N-2}$ and $V_{N-2}$. Use this $U_{N-2}$ and $V_{N-2}$ to correct the networks. Continue this process until both the networks converge. These networks yield optimal $U_{N-2}$ and $V_{N-2}$.

6. Using random $Z_{N-2}$ into $U_{N-2}$ network obtains optimal $U_{N-2}$. Similarly obtain optimal $V_{N-2}$. Use $Z_{N-2}, U_{N-2}$ and $V_{N-2}$ to obtain $Z_{N-1}$ and input to $\lambda_{N-1}$ network to generate $\lambda_{N-1}$. Use $Z_{N-2}, U_{N-2}, V_{N-2}$ and $\lambda_{N-1}$ in costate equation to obtain optimal $\lambda_{N-2}$. Train $\lambda_{N-2}$ network with $Z_{N-2}$ as input. We have $\lambda_{N-2}$ network that yields optimal $\lambda_{N-2}$.

7. Repeat steps 5 and 6 with $K=N-1, N-2, \ldots, 0$, until we get $U_0, V_0$.

A schematic of the network development is presented in Fig.1

![Figure 1 Schematic of Adaptive-Critic training](image-url)
5. RESULTS AND DISCUSSION

Preliminary results have been obtained for the problem using the standard procedures. Further analysis will be done using the adaptive critic approach. The current results will help in validating those obtained from adaptive critic. The pursuit-evasion problem will be studied for different and more complex strategies of evader and that of the pursuer. These results would be presented at the conference.

Figure 2. Distance between Pursuer and Evader with Time
Figure 3. Control of Pursuer and Evader Vs time
6. REFERENCES


