A Genetic Programming Approach to Strategy Optimization in the Extended Two-Dimensional Pursuer/Evader Problem

Frank W. Moore
303 Russ Engineering Center
Wright State University
Dayton, OH 45435
fmoore@kiwi.cs.wright.edu

Dr. Oscar N. Garcia
303 Russ Engineering Center
Wright State University
Dayton, OH 45435
ogarcia@valhalla.cs.wright.edu

ABSTRACT

This paper describes a genetic programming system that evolves optimized solutions to the extended two-dimensional pursuer/evader problem. The pursuer/evader problem is a competitive zero-sum game in which an evader attempts to perform maneuvers to escape a faster, more agile pursuer. The extended problem is more realistic than previous formulations because the evader and pursuer are modeled as point masses that are capable of limited thrusting and turning forces, and are subject to drag forces and momentum. The pursuer initially aims at a predicted capture point, and uses proportional navigation to attempt to maintain a constant line-of-sight angle with the evader. The game ends favorably for the evader if it manages to stay outside the lethal radius of the pursuer for the duration of the encounter (limited by the effective range of the pursuer). To solve the extended two-dimensional pursuer/evader problem, a strategy must be identified by which an evader (such as an F-16C fighter aircraft) may maneuver to successfully evade pursuers (such as surface-to-air missiles) starting from a wide range of potentially lethal relative initial positions.

1. Introduction

The two-dimensional pursuer/evader problem (2DPE) is a competitive zero-sum game in which a faster, more agile pursuer is given a limited amount of time to capture an evader as both are traveling across a plane (Hamalainen and Ehtamo, 1990). The game ends favorably for the evader if it manages to stay outside the *lethal radius* of the pursuer (the maximum distance at which a capture is considered to have occurred) for the duration of the encounter. A solution to the 2DPE problem must incorporate an optimized strategy for maneuvering the evader in a manner that successfully escapes the pursuer, regardless of the initial conditions of the system. While several game-theoretic and analytical solutions to pursuer/evader problems exist (Basar and Bernhard, 1989; Krasovskii and Subbotin, 1988; Zarchan, 1990), many of these solutions impose unrealistic constraints on pursuer and evader motions; other solutions are unrealistic because they assume that the pursuer and evader are massless points capable of instantaneous changes in direction; many solutions are unrealistic because they do not include such factors as drag forces and momentum.

(Koza, 1992) described how genetic programming (GP) was used to obtain an optimized solution to a simple 2DPE problem. In this formulation, Koza assumed that both pursuer P and evader E were *massless points* traveling in a plane at constant speed, with $v_p > v_e$, and that P and E were capable of instantaneous changes in direction. Given these conditions, the optimal strategy for the evader was to travel in a straight line away from the

pursuer. Koza described how the fitness-proportionate reproduction and crossover mechanisms of GP automatically evolved an evader control program that demonstrated near-optimal performance. The assumptions of constant speed, zero mass, and instantaneous changes in direction reduced the complexity of Koza's 2DPE problem.

2. Problem Formulation

The *extended two-dimensional pursuer/evader problem* (E2DPE) defined in this paper models pursuer P and evader E as point masses (rather than massless points) whose motions across a plane are controlled by thrusting forces (applied in the direction of the velocity vector) and turning forces (applied in a direction that is perpendicular to the velocity vector). Both P and E are affected by drag forces and momentum; instantaneous changes in direction are not possible. Since acceleration (a), force (F), and mass (m) are related by the equation a = F/m, the acceleration of P or E depends on its current mass, as well as the magnitude of the applied thrusting and turning forces. The effect of these forces depends on the current state (the position, velocity, and acceleration vectors) of P or E. The maximum distance over which P can pursue E depends on the type of pursuer. P captures E as soon as the distance between them becomes less than a pursuer-specific lethal radius. The evader performs maneuvers by executing specific combinations of thrusting and turning forces in specific sequences. The optimal strategy for the evader is to maneuver in a manner that maximizes the likelihood of evading the pursuer.

The initial conditions of the E2DPE problem are illustrated in Figure 1. Prior to the start of the encounter, the pursuer uses the initial state (position, velocity, and acceleration) of the evader to predict an intercept point, and launches itself at maximum thrust towards that point. If the evader fails to maneuver, the pursuer will capture the evader at (or very close to) this intercept point. To compensate for evader maneuvers, the pursuer will rely on the highly effective *proportional navigation* technique (Ball, 1985). Proportional navigation causes the pursuer to accelerate in the direction perpendicular to the instantaneous line-of-sight from the pursuer to the evader; the magnitude of this acceleration is calculated as follows:

$$n_c = N' \, V_c \, (d\lambda/dt)$$

where N' is a unitless designer-chosen gain known as the *effective navigation ratio*, and V_c is the pursuer-evader closing velocity vector (the negative rate of change of the distance from the pursuer to the evader). The time derivative of the line-of-sight angle λ is known as the *line-of-sight rate*. The effective navigation ratio for a given guidance system may be determined mathematically from a complex series of computations (Barron, 1995); for practical guidance systems, optimal values for N' range between 3 and 5 (Ramo and Pucket, 1959).

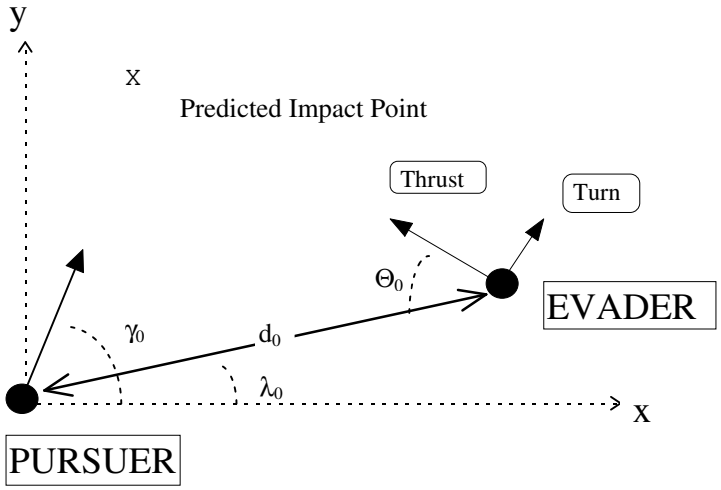

Figure 1. The Extended Two-Dimensional Pursuer/Evader Problem (Initial Conditions).

3. Genetic Programming Implementation

A genetic programming solution to the E2DPE problem was successfully implemented during this research. During each generation of the genetic programming approach, each member of a population of programs for maneuvering the evader was trained against each member of a training population of pursuers. Each GP run optimized maneuvers against a single type of pursuer. Proportional navigation was used by all pursuers. The pursuer and evader had complete knowledge of each other's current state (relative position, velocity, and acceleration). A fitness function was used to estimate the survivability of each evasion program during each simulated encounter; the aggregate fitness of a particular program reflected its fitness when independently evaluated against all of the pursuers in the training population. Fitness-proportionate reproduction (20%), together with crossover (80%), was used to create each new generation. Mutation was not used. Each GP run was terminated after a fixed number of generations.

An optimized evasion program considers the current state of the evader and pursuer, and outputs commands that assert thrusting and turning forces at appropriate moments:

THRUST -- Sets the thrusting force to the specified percentage of the evader's maximum thrust.

TURN – Enacts a turning force equal to the specified percentage of the evader's maximum turning force; the force acts in a direction perpendicular to the evader's current velocity vector; negative values indicate that the turn should be to the left, while positive values cause a turn to the right.

The best-of-run program represents an optimized evasion technique for a specific type of evader and a specific type of pursuer.

The objective of the evader was to survive pursuer attack, regardless of the initial state of the evader and the relative launch position of the pursuer. The initial position of the evader relative to the pursuer at launch time was defined by a fixed set of positions located within the effective engagement space of the pursuer. Note that by a simple rotation of the reference coordinate system (fixed at the launch site of the pursuer), the initial pursuer/evader line-of-sight angle λ could be considered to be the same for all pursuer/evader pairs. For this reason, the following variables were sufficient for describing the initial configuration of each confrontation:

(a) The *line-of-sight distance* between the evader and the pursuer;
(b) The *velocity vector* of the evader at the time the pursuer is launched.

For each pursuer type, the minimum and maximum effective range of the pursuer defined the range of possible initial line-of-sight distances. The line-of-sight distance and the initial velocity vector of the evader were used by the pursuer to predict an intercept point. The pursuer was then launched in the direction of the intercept point, and changed its trajectory only in response to evader maneuvers (Zarchan, 1990). The aggregate fitness of a specific program reflected its fitness when trained against each of these initial pursuer positions.

A tableau for the E2DPE problem is shown Figure 2. Cartesian coordinates were used to represent components of position, velocity, and acceleration for both the evader and the pursuer. This simple set of terminals proved to be sufficient to allow the GP system to converge to a solution of this problem. This formulation fixed the origin of the coordinate system ($x = 0$, $y = 0$) at the position of the pursuer. PX and PY thus continually designated the relative displacement from the pursuer to the evader in two dimensions.

For each of these functions, an "argument" may consist of any syntactically valid composition of functions, variables, and constants that returns a floating-point value in the range [-1.00 … +1.00]. Function ifPX is a two-argument selection function: if the x-displacement of the evader relative to the position of the pursuer is negative, then the first argument is evaluated; otherwise, the second argument is evaluated. Function ifPY is a two-argument selection function defined in a similar manner for the y-displacement of the evader. Function ifDistance is a three-argument selection function: if the current distance between the pursuer and evader is less than the percentage of the pursuer's maximum pursuit distance specified by the absolute value of the first argument, then the second argument is evaluated; otherwise the third argument is evaluated. Each of these functions returns the value of the evaluated argument.

Functions setThrust and hardTurn are single-argument functions. Function setThrust causes the thrust output of the evader to be set to the percentage of its maximum thrust specified by its argument; for example, setThrust (0.9) will set evader thrust to 90% of its maximum possible value. SetThrust ignores the sign of its argument; thrust always acts in the direction of the current evader velocity vector. Function hardTurn causes the evader to execute a turn whose g-force equals the percentage of the maximum allowable turning force of the evader/pilot system specified by its argument; for example, if the maximum turning force is 4 gravities (g's), then hardTurn (0.5) will cause the evader to execute a 2g turn in a direction which is perpendicular and to the right of the current evader velocity vector; the function call hardTurn (-0.5) would result in a 2g evader turn to the left. Both setThrust and hardTurn are assumed to act instantaneously. Both of these functions return the value of their input argument.

Each fitness case is denoted by values from specified finite sets J and K. The first value, J_m, identifies the initial line-of-sight distance from the pursuer to the evader. If D_{min} and D_{max} denote the minimum and maximum effective launch distances for the pursuer, then the initial line-of-sight distance d_0 may be calculated as follows:

$$d_0 = D_{min} + (J_m * (D_{max} - D_{min}))$$

D_{min} and D_{max} depend upon the type of pursuer.

The second value, denoted K_n, identifies the angle that the initial velocity vector of the incoming evader makes with the line-of-sight

Objective:	Determine an optimized evasion strategy for the extended two-dimensional pursuer/evader problem.
Terminal Set:	PX, the displacement in the x-direction from the pursuer to the evader. PY, the displacement in the y-direction from the pursuer to the evader. ***R***, the ephemeral random floating-point constant ranging from -1.0 to 1.0.
Function Set:	ifPX ifPY ifDistance setThrust hardTurn
Fitness Cases:	Numerous fitness cases which differ according to the distance from the pursuer to the evader at the start of the encounter, as well as the acute angle between the initial velocity vector of the evader and the line-of-sight vector from evader to pursuer.
Raw Fitness:	The number of times the distance between the pursuer and evader is less than or equal to the lethal envelope of the pursuer.
Standardized Fitness:	Same as Raw Fitness for this problem.
Hits:	The number of fitness cases that result in capture of the evader prior to the maximum pursuit time of the pursuer. (The encounter is also terminated if the distance between the evader and the pursuer exceeds a pursuer-specific value, at which time the pursuer is considered to have missed the evader.)
Wrapper:	N/A
Parameters:	Population Size M = 100, Maximum Number of Generations G = 21.
Success Predicate:	None.

Figure 2. A Tableau for the E2DPE Problem

from the evader to the pursuer. Let Θ_0 denote this angle. If Θ_{min} and Θ_{max} denote the minimum and maximum initial value of Θ, then Θ_0 may be calculated in the following manner:

$$\Theta_0 = \Theta_{min} + (K_n * (\Theta_{max} - \Theta_{min}))$$

To maintain the relative geometry illustrated in Figure 1 for the pursuer/evader problems addressed by this research, Θ_{min} and Θ_{max} described a range of values between 10 and 80 degrees. For this study, the magnitude of the evader's velocity vector (its "speed") at pursuer launch time was assumed to be the same for all

encounters. Thus, each fitness case corresponded to a specific combination of J_m and K_n.

The goal of this research was to identify a methodology for evolving optimized evasion strategies (in the form of programs) for a variety of pursuer populations. We have begun with the two-dimensional problem to reduce the required amount of computation as much as possible. Implicit in the model described above is the assumption that the magnitude of the turning force is independent of the velocity of the evader. Additionally, limitations on the maximum sustainable g-force of the evader were imposed by restricting the magnitude of the evader's turning force; the contribution of thrust to current evader g-force was ignored. We also assumed that the evader stalled if its speed fell below a specified minimum value, resulting in a "kill" for the pursuer. For the purposes of this study, all evaders were assumed to be traveling *inbound* (towards the pursuer) at the start of each confrontation, with lead angle $\gamma_0 > \lambda_0$, as shown in Figure 1.

4. Training

The GP system implemented for this research incorporated physical data (including mass) and performance characteristics (such as maximum thrust, maximum turning rate, fuel consumption rate, and drag coefficients) of an F-16C aircraft evader (Lambert and Munson, 1994) and three types of surface-to-air missile (SAM) pursuers: the SA-6, SA-13, and SA-15 (Cullen and Foss, 1995). An effective navigation ratio N' = 3 was assumed for each pursuer. For each type of pursuer, fifteen best-of-run programs were evolved using training populations of pursuers defined by the cross-product of sets J and K, as follows:

Group 1, Training Runs 1-5:
$J_m \in \{0.3, 0.45, 0.55, 0.7\}$
$K_n \in \{2/9, 3/9, 4/9, 5/9\}$
Group 2, Training Runs 6-10:
$J_m \in \{0.45, 0.5, 0.55, 0.65\}$
$K_n \in \{2/9, 3/9, 4/9, ½, 5/9, 6/9, 7/9, 8/9\}$
Group 3, Training Runs 11-15:
$J_m \in \{0.2, 0.3, 0.4, 0.45, 0.5, 0.6, 0.7, 0.8\}$
$K_n \in \{2/9, 3/9, 4/9, 5/9\}$

Among each of these three groups, individual runs differed in the random number seed used (1.0, 0.8, 0.6, 0.4, or 0.2). Each run used a program population size M = 100, and ran for a maximum of G = 21 generations. Fitness was determined by independently evaluating the survivability of each program against each element of a training population of pursuers launched from the potentially lethal positions defined by the cross-product of sets J and K.

During each of Training Runs 1-10 and for each type of pursuer, our GP system was able to evolve a best-of-run program that implemented an evasion strategy that survived attack from 100% of the pursuers in the training population. In contrast, *none* of Training Runs 11-15 was able to evolve a program capable of defeating SA-13 or SA-15 pursuers identified by $J_m = 0.2$, $K_n = 2/9$. These results are directly analogous to actual aircraft/SAM encounters: in the real world, the most lethal SAMs are those launched at close range with small lead angle.

5. Test Results

After evolving optimized solutions to all 45 problems, each best-of-run program was independently tested against a large, representative population of 128 pursuers described by the cross-product of the following J and K values:

$J_t \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8\}$
$K_t \in \{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85\}$

Note that the test population defined by sets J_t and K_t has a much finer granularity than the training population, and describes a completely different set of pursuer/evader encounters. The results of these tests are tabulated in Figure 3. Each value represents the number of pursuers successfully evaded by the corresponding best-of-run program during subsequent testing.

6. Conclusions

The results of this research were encouraging. By incorporating forces, masses, and drag, the E2DPE problem is much more realistic than other pursuer/evader problems described in the available literature. The ability to evolve best-of-run evasion programs that successfully optimize survivability against a representative test population of pursuers lends credibility to and broadens the applicability of the GP methodology. The system described in this paper is being continuously refined to more

accurately model actual aircraft evaders and SAM pursuers. It is also being extended to investigate methods of using GP to optimize missile countermeasures under conditions of uncertainty about the type and/or state of the SAM; to include the evader's use of electronic countermeasures (including chaff, flares, and jamming); and to solve the three-dimensional pursuer/evader problem.

VI. BIBLIOGRAPHY

Ball, R. E., 1985. *The Fundamentals of Aircraft Combat Survivability Analysis and Design*, AIAA Education Series, AIAA Inc.

Barron, R. L., 1995. "Reduced-Computation End-Game Steering Laws for Predictive Guidance", in *Journal of Guidance, Control, and Dynamics, Vol. 18 No 2, March-April 1995*, pp. 306-315, American Institute of Aeronautics and Astronautics, Inc.

Basar, T. S. and P. Bernhard (eds.), 1989. *Differential Games and Applications*, Lecture Notes in Control and Information Sciences Vol. 119, Springer-Verlag.

Cullen, T. and C. Foss, 1995. *Jane's Land-Based Air Defence: 1995-1996*, Jane's Information Group, Inc.

Hamalainen, R. P. and H. K. Ehtamo (eds.), 1990. *Differential Games -- Developments in Modeling and Computation*, Lecture Notes in Control and Information Sciences Vol. 156, Springer-Verlag.

Koza, J. R., 1992. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press.

Krasovskii, N. N. and A. I. Subbotin, 1988. *Game-Theoretical Control Problems*, Springer- Verlag.

Lambert, M. and K. Munson (eds.), 1994. *Jane's All the World's Aircraft: 1994-1995*, Jane's Information Group, Inc.

Ramo, S. and A. Pucket, 1959. *Guided Missile Engineering*, McGraw-Hill, pp. 176-180.

Zarchan, P., 1990. *Tactical and Strategic Missile Guidance*, Progress in Astronautics and Aeronautics Vol. 124, American Institute of Aeronautics and Astronautics.

Best-of-Run Program	vs. SA-6s	vs. SA-13s	vs. SA15s
from Training Set 1	96 (75.0%)	115 (89.8%)	110 (85.9%)
from Training Set 2	127 (99.2%)	113 (88.3%)	109 (85.2%)
from Training Set 3	112 (87.5%)	113 (88.3%)	106 (82.8%)
from Training Set 4	127 (99.2%)	113 (88.3%)	100 (78.1%)
from Training Set 5	95 (74.2%)	113 (88.3%)	111 (86.7%)
from Training Set 6	127 (99.2%)	116 (90.6%)	114 (89.1%)
from Training Set 7	127 (99.2%)	113 (88.3%)	90 (70.3%)
from Training Set 8	112 (87.5%)	110 (85.9%)	106 (82.8%)
from Training Set 9	127 (99.2%)	113 (88.3%)	87 (68.0%)
from Training Set 10	127 (99.2%)	113 (88.3%)	111 (86.7%)
from Training Set 11	125 (97.7%)	116 (90.6%)	111 (86.7%)
from Training Set 12	128 (100%)	115 (89.8%)	113 (88.3%)
from Training Set 13	128 (100%)	120 (93.8%)	112 (87.5%)
from Training Set 14	127 (99.2%)	115 (89.8%)	113 (88.3%)
from Training Set 15	128 (100%)	120 (93.8%)	114 (89.1%)

Figure 3. Test Results for Best-of-Run Programs for Each Pursuer Type