JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 38, No. 1, January–February 2001

Design of a Guided Missile Interceptor Using a Genetic Algorithm

M. B. Anderson*
Sverdrup Technology, Inc., Eglin Air Force Base, Florida 32542
and
J. E. Burkhalter† and R. M. Jenkins‡
Auburn University, Auburn, Alabama 36830

An apportioned pareto genetic algorithm was used to manipulate a solid rocket design code, an aerodynamic design code, and a three-loop autopilot to produce guided missile interceptor designs capable of accurately engaging a high-speed/high-altitude target. Definition of the optimization problem required 29 design variables, and 4 primary goals were established to assess the performance of the interceptor designs. Design goals included the following: minimize miss distance, minimize intercept time, minimize takeoff weight, and minimize maximum *g* loading. In 50 generations, the genetic algorithm was able to develop 2 basic types of external aerodynamic designs that performed nearly the same, with miss distances less than 1.0 ft. The solid rocket motors that propelled these external shapes shared common characteristics such as a large initial burning area and a large combustion chamber volume. The genetic algorithm did not prefer maximizing the amount of fuel within the rocket motor case (high fuel volume ratio). A higher fuel volume ratio typically means higher launch weight, but does not necessarily guarantee faster intercepts given finite thermal limits. Examination of the intercept trajectories themselves shows that standard proportional navigation guidance works adequately, but could probably be improved by thrust compensation, especially during the launch transient. The three-loop autopilot performs well even for high-altitude engagements, and the analytic gain determination makes the autopilot straightforward to implement.

Nomenclature

b_t	=	exposed semispan of tail, in.
b_w	=	exposed semispan of wing, in.
C_{rt}	=	tail root chord, in.
C_{rw}	=	wing root chord, in.
D_i	=	point score of each goal i
D^*	=	diameter of the throat, in.
d_s	=	domination strength
f_r	=	fillet radius, in.
g_r	=	goal rank order
L_{nose}	=	nose length, in.
L_{tot}	=	total body length excluding nozzle, in.
Nose	=	shape of nose (1, ogive, and 2, cone)
N_{st}	=	number of star points
N'	=	effective navigation ratio or gain
n_g	=	total number of goals
R_{bi}	=	grain outer radius and case inner radius, in.
R_{body}	=	missile body radius, in.
R_{exp}	=	nozzle expansion ratio
R_i	=	inner star radius, in.
R_p	=	outer star radius, in.
TR_t	=	tail taper ratio, C_t/C_r
TR_w	=	wing taper ratio, C_t/C_r
X_{let}	=	distance from nose tip to tail leading edge, in.
X_{lew}	=	distance from nose tip to wing leading edge, in.
x_{gl}	=	grain length, in.
ε	=	angular fraction of star point, rad
ζ	=	damping ratio
Θ	=	Euler vertical launch angle, deg
λ_{tet}	=	tail trailing-edge sweep angle, deg
λ_{tew}	=	wing trailing-edge sweep angle, deg
τ	=	time constant, s
ψ	=	Euler launch heading angle, deg
ω_{cr}	=	crossover frequency, rad/s

Received 8 February 2000; revision received 1 June 2000; accepted for publication 16 June 2000.

*Deputy Director, Development Test Department. Senior Member AIAA.
†Professor, Aerospace Engineering. Associate Fellow AIAA.
‡Associate Professor, Aerospace Engineering. Senior Member AIAA.

Introduction

WITH the addition of guidance, autopilot, and an airframe with movable control surfaces, basic rocketry expands into a more lethal and much more precise means of waging war. Rather than increasing the size of the warhead being delivered (to make up for a loss in delivery accuracy), modern weaponeering has tended to use a small warhead coupled with an accurate control system. When a system has a guidance system and an autopilot, there is a tendency to compensate for less than stellar aerodynamic designs by shifting more of the delivery problems over to the autopilot. As a result, autopilots are typically very good and very robust, but the airframe and aerodynamics of the overall system are almost an afterthought. Overall system performance and system capability, therefore, suffers because of the overreliance on the autopilot to compensate for weaknesses in the aerodynamic performance of the weapon system. The goal of this research is to let an intelligent systems tool, a genetic algorithm, design the aerodynamic shape while at the same time designing the propulsion system and key autopilot variables. This all-at-once approach to missile design is intended to provide a system capable of producing good aerodynamic shapes in addition to the good performance expected from an autopilot.

Genetic Algorithms

Genetic algorithms (GAs) are considered to be an intelligent systems tool, like artificial neural networks, fuzzy logic, and knowledge-based systems. GAs are so called because they attempt to use the supposition of evolution as a basic mechanism for improvement, that is, learning/survival of the fittest, in solving a problem. All GA work stems from the pioneering efforts of Holland, whose classic book[1] set the foundation for population-based adaptive optimizers. Following the terminology of true genetics researchers, the computational GAs developed by Holland (and his students) encode potential solutions into chromosomelike structures, then allow these structures to compete, reproduce, and mutate to produce (hopefully) better solutions over time. GAs have been increasingly used in optimization studies over the past decade and have more recently been used in multidisciplinary optimization. There should be an emphasis on improvement rather than optimization in a multidisciplinary context, simply because for complex problems it is not possible to prove that an optimum has been found.

GA Terminology and Operations

GA terminology and operations mimic the terminology in genetics and biology. Those terms pertinent to understanding the GA descriptions in this document are defined here. Naturally, some of these definitions are couched in optimization terms because optimization is such a pertinent field of application of GAs:

1) A population is a collection of competing potential solutions to a problem.

2) A generation is a population at a certain time (iteration).

3) Encoding is the process by which real numbers are converted into their binary (in this research) representation.

4) An alphabet relates to the encoding scheme that is being used having a certain number of values that are required to represent the encoding scheme. For example, a binary alphabet can take on two values only (1 or 0), whereas an English alphabet would have 26 possible values. A standard binary alphabet was used in this research.

5) Chromosomes are the encoded parameters in a potential solution in concatenated form.

6) Individual genes make up chromosomes; in a binary alphabet these are frequently called bits.

7) An allele is the value of a particular gene (1 or 0 for binary).

8) The objective or fitness function is the function that determines the performance of a particular chromosome. This is the function being optimized through parameter manipulation.

9) Reproduction is the process by which new generations are formed.

10) Selection is the process through which survivors are selected. For this research tournament selection was used. The best of every two competing solutions is allowed to survive to reproduce.

11) Mating is the mechanism by which two surviving chromosomes can exchange binary information (via crossover) and produce offspring.

12) Crossover is the actual exchange of genes/bits between two survivors. Single-point crossover, which was used in this research, means that two sets of chromosomes are broken apart at a random location and spliced back together such that genes beyond the break point are exchanged between the two chromosomes.

13) Mutation is a random process where a single bit changes its value from 0 to 1 or 1 to 0.

Many facets control the way that a GA works. A potential solution first has to be encoded, along with all of the other potential solutions that form a generation. This population is then fed one at a time to the objective function so that a measure of the performance of each member of the population can be ascertained. The better performers then have a higher probability of surviving the tournament selection process to reproduce the next generation. Mating between survivors, with its related crossover function, is the mechanism by which new populations are formed. Mutation is also allowed to occur and helps preserve genetic diversity. Over time, as generations build on the successes of previous generations, the performance of the entire population increases as the algorithm learns what allele values produce good answers. Poor performers die off (lower reproduction rate), and over many generations the best performers split and recombine with each other to produce even better solutions. Many researchers tout GAs as global optimizers, and that is true because of mutation and the general probabilistic nongradient nature of GAs; however, in engineering applications good answers (answers that will suffice) are desired as fast as possible. Whether the answer is the absolute global optimum or not is less of a concern than whether a good answer can be found in the time allotted by management. Besides, for complicated problems there is no analytical way to determine the global optimum, and so arguments over whether the true optimum has been found are academic.

Contrasts with Conventional Optimization Approaches

GAs are population based. Rather than starting from a single best guess and then marching toward (gradient, conjugate gradient, etc.) a local optima based on deterministic transition rules, that is, sensitivity derivatives, GAs test multiple solutions (population) and base the next population on strictly probabilistic transition rules. Because a GA does not work based on gradients or derivatives in the objective function based on the parameters, they can work quite easily on irregular functions such as step functions and discrete disjointed functions. They can also work on problems with multiple local optima and yet not get caught in these local traps.

Pareto Optimality

Pareto GAs[2] differ from standard GAs in that they operate on multiple goals or objectives simultaneously rather than having one fitness function. The goals are optimized individually to determine parameter sets that work well for each goal. Through the mating of good parameter sets for each individual goal, a family of parameter sets that work well across the spectrum of goals is obtained. This family of solutions is called the pareto optimal (p-optimal) set for the multiobjective/multigoal problem. The operation of the pareto scheme in the GA software is by domination. One goal set, defined as the collection of individual goal performances based on one parameter set, must clearly dominate another if the parameter set is to survive. Goal set $\boldsymbol{A}$ dominates set $\boldsymbol{B}$ if the following two statements are true:

$$\forall_i (\boldsymbol{A}_i \geq \boldsymbol{B}_i), \qquad \exists_i (\boldsymbol{A}_i > \boldsymbol{B}_i)$$

When two members of the population are chosen for the tournament selection procedure, the domination rules are examined to see whether one member dominates the other. The clear winner is retained for the next generation. If there is no clear winner, a situation called nondomination, the winner is selected at random for survival.

Apportioned GA

The apportioned pareto algorithm goes one step beyond the standard pareto algorithm to address the issue of solutions that work well in all goal areas: it considers the relative importance of the goals. Essentially, this algorithm is an automated weighting procedure that does not require grouping multiple objectives into a single fitness function. Consider the problem of aircraft design, where range would eagerly be sacrificed for safety, but it is still important to get acceptable range. This tradeoff is where the ranking consideration comes into play. In each goal area a winner-takes-all strategy is used to determine the total number of points that belong to goal set $\boldsymbol{A}$ and goal set $\boldsymbol{B}$. After each goal area is compared, the goal set with the most points wins, and its chromosome survives the selection process. The domination strength parameter essentially dictates the distribution (apportionment) of scoring points among the goals according to the function

$$D_i = \exp\{f[1 - (g_r/n_g)]\} \tag{1}$$

where

$$f = \frac{\ln(d_s)}{1 - (1/n_g)} \tag{2}$$

The term apportionment implies that there is a finite amount of something that must be distributed in some fashion. The apportioning GA follows the artificial life studies of von Neuman,[3] Farmer et al.,[4] and Langton.[5] In these artificial life studies, competing life units strive to possess the life-giving food that they need for survival. There is only a finite amount of food available, however, and so the life units that obtain the food are able to survive. The simplicity of this approach is that there is only one parameter d_s that establishes how important the goals are relative to each other. When d_s is large, the highest priority goal can dominate the solution to such an extent that it would take superior performance in every other goal area to outweigh this one goal. When d_s is small, the goals are more equally weighted. The lower limiting value, when d_s is 1.0, yields a goal-counting scheme. The highest ranking goal is functionally mapped so that the number of points that can be garnered through exhibiting the best performance in this area is the domination strength value. Likewise, the lowest ranking goal always garners one point, regardless of d_s and the total number of goals. Because each goal is assigned a certain number of points based on the domination strength, the magnitude of domination strength determines how many other goals at given rankings must team together to outweigh the importance of goal 1. Low dominance strengths usually mean that any two other goals can outweigh goal 1 (unless goal 1 also has a partner), and high dominance strengths means that it might take all of the other goals acting in unison to unseat goal 1.

Previous Work

Rather than discuss the historical development of GAs or the simulation components used in this research, it is more pertinent to discuss previous applications of GAs to autopilot and control systems. Norris and Crossley[6] recently used a GA to find gains to control a standard pedagogical two-disk torsional spring system. The approach used a very simple two-loop proportional–integral control system with velocity feedback. The two-loop controller had three gains that needed to be determined as a function of variable spring stiffness. The objective functions were defined such that good gain values would produce little error between the commanded and achieved disk rotation angles. Because two separate disks were being commanded in twist, a pareto GA was used to try to minimize the rotational errors in both disks simultaneously. At the conclusion of 80 generations, the resulting family, that is, population of 80 members, of three controller gains were hybrid performers that would work reasonably well for both disks.

Another interesting GA controller application was recently presented by McGookin et al.[7] to control the steering of oil tankers to multiple waypoints in a narrow channel. In their work, the control system consisted of a two-loop autopilot and a sliding mode controller. The four parameters being optimized consisted of the first and second heading loop poles (frequency plane), a heading switch gain, and a so-called heading boundary-layer thickness. The goal of the study was to find values for these four parameters such that the oil tanker passed within an acceptable distance of each waypoint while minimizing rudder movement. Minimizing rudder movement saves fuel and time. The results shown indicate that the GA found excellent values of the four gains in 100 generations. Rudder movement was minimal, and each waypoint was reached with very little error. In terms of the GA operation in this study, it is interesting that McGookin et al.[7] used a 5% mutation rate, which is at least an order of magnitude higher than conventional values of this parameter. Because there is no agreement among GA researchers about value ranges for crossover and mutation, it is interesting to note what values other researchers use for their applications.

Another study by Martin[8] addressed the issue of autopilot gain scheduling by using GAs to replace the ad hoc design process typical in linear gain scheduling with a genetically fit hyperplane-surface strategy. The GA was basically used to optimally design the gain schedule. Of course a gain scheduling approach that might work for one scenario could be inadequate without adaptation to other scenarios. Adaptation strategies have been pursued by Karr and Harper[9] using GAs to augment fuzzy logic controllers. Coupling GAs with neural networks appears to offer improvement to adaptive control that neither approach has independently. The fuzzy controller (neural network) uses a rule-of-thumb strategy to control a chemical system, but the chemical system is periodically changed, thereby invalidating some of the rules of thumb. As the system changes, a learning algorithm (the GA in this case) tests new rules of thumb so that the fuzzy controller can continue to control the chemical system. For autonomous systems,[10] this type of approach has obvious advantages assuming the GA can keep up with the rate of change of the system. Karr et al.[11] later expanded this work to control the rendezvous of two spacecraft. Another excellent work in adaptive fuzzy logic controller design using GAs was done by Homaifar and McCormick[12] to control a simple electronic cart. The GA designed both the rules of thumb and the membership functions for the system in an automated process that did not require human input.

Two of the more modern approaches that have demonstrated applicability to solid rocket motor (SRM) design were done by Clergen[13] and McCain.[14] Clergen[13] developed a computerized expert system with a hypertext user interface to aid designers in selecting preliminary designs based on past experience, that is, the experts. With this system the designer can select, with a user-friendly interface, design criteria such as minimum motor mass and obtain preliminary design parameters that might be suitable for the mission being planned. This system is built around a database of known systems. McCain's[14] system is also an expert system in the sense that it is heuristic. The heuristic performs an independent design variable selection, and these design variables are then passed to an existing pattern search optimization package to develop rocket performance characteristics. The heuristic then analyzes the effect of altering each independent SRM design variable and selects, based on sensitivity/partial derivatives, the variables to alter for the next design attempt.

System Components

The guidance algorithm used in this study is standard proportional navigation (called ProNav). ProNav requires the specification of one gain value to govern how aggressively heading errors are taken out of the system. According to Zarchan,[15] typical ranges for this parameter are 3–5 (unitless) for tactical weapon systems; thus, given this variation, the effective navigation ratio is a variable that the GA can determine.

The autopilot chosen for this study is the so-called three-loop pitch/yaw autopilot. This autopilot design was chosen because of its simplicity, because it is actively being used in several existing weapon systems, and because it is possible to calculate analytically the proper system gains (for all flight conditions) based on a few specified autopilot performance parameters.

The aerodynamic prediction methodology chosen for this study was developed by Washington.[16] The software package, called AeroDesign, is a semi-empirical tool based on large wind-tunnel databases for various wing/body/tail geometries. AeroDesign provides body axis system force and moment coefficients at a wide variety of flight conditions, from subsonic Mach numbers to supersonic Mach numbers and from low angles of attack to nearly a 20-deg angle of attack. AeroDesign was modified to include two axial force considerations that were not part of the original software. First, the fineness ratio of the nose of the missile is compared to a Sears–Haack (see Ref. 17) body, and if the nose is not slender enough, a drag penalty proportional to the nose bluntness is added to the baseline axial force coefficient. The second axial force coefficient correction was implemented to correct for cases where the rocket nozzle exit diameter actually exceeds the diameter of the body.

AeroDesign was further modified to provide aerodynamic damping derivative estimates and linear aerodynamic coefficient contributions for deflected control surfaces in the pitch and yaw planes in a format and flight condition range compatible with a guided six-degree-of-freedom (DOF) simulation.

The solid rocket performance software used in this study is an erosive burning star grain design program that is suitable for preliminary design studies. The basic software was developed by Burkhalter[18] and Sforzini[19] for use in an advanced undergraduate/graduate course at Auburn University.

The six-DOF equations of motion were obtained from Etkin.[20] Quaternions were used rather than Euler angles to avoid gimbal lock. The conventional body-fixed acceleration and moment equations were modified to include aerodynamic damping terms and contributions due to control surface deflections. For this study, a tail-control missile was assumed, but there is no reason why a canard-control system could not also be analyzed.

Assumptions

For this study, it was assumed the missile is a rigid body with insignificant cross products of inertia. It was also assumed that the inertial measurement unit (IMU) was perfect, meaning that the actual and measured accelerations and rates are identical. For simplicity, it is also assumed that the perfectly measured accelerations include appropriate translations to compensate for center of gravity movement as fuel is expended. In real systems there will be lag and measurement error in the IMU, but for this preliminary design study these simplifications are appropriate. The actuator was modeled as a second-order system[15] with a damping of 0.7 and a natural frequency of 125 rad/s. These simplifications mean that the system is fifth order if the airframe is considered to be a second-order response.

There are some fundamental assumptions made in the formulation of the propulsion software that make it suitable for preliminary design studies. First, the pressure varies throughout the chamber. However, the pressure is calculated only at the head end, P_1, and at the grain end, P_2. The chamber pressure P_{ch} is then defined as the average of these two pressures. Second, the burn rate of the propellant also varies over the entire surface and is subject to erosive burning. The burning rate at any point can be defined as $r = aP_{ch}^n(1 + k \cdot V)$, where k, a, and n are burning rate constants that have been experimentally determined for the fuel. At the head end of the grain $V_1 = 0$, and at the aft end of the grain the velocity equals V_2. The burning rate calculation uses the average velocity of these two locations. Third, the grain burns normal to the centerline of the rocket, that is, no end burning. Fourth, the flow is isentropic between the aft end of the grain and the throat. Fifth, the flow obeys the perfect gas law. Sixth, the chamber pressure varies with time, but is essentially constant during the discharge of a single particle. Seventh, the flow is one-dimensional and steady. Eighth, there is no deformation of the propellant due to acceleration, pressure, or viscous forces. Ninth, the temperature is uniform throughout the grain, but the grain is temperature sensitive.

The rocket motor to be designed by the GA has certain definable characteristics, such as the strength of the combustion chamber material, which should be known before the design process begins. For this study, the following rocket characteristics were used:

1) The propellant is ammonium perchlorate (80%).
2) The initial temperature of the propellant is 68°F.
3) The design chamber pressure (i.e., maximum chamber pressure for case structural design) is 3000 psi. Given this chamber pressure, case thickness is determined by using a factor of safety of 1.5.
4) The allowable stress in the case is 195,000 psi.
5) The factor of safety is 1.5.
6) The case is made of a steel alloy with a density of 0.28 lbm/in^3.
7) The nozzle is made of an aluminum alloy with a density of 0.19 lbm/in^3.

For this study ammonium perchlorate (80%) was chosen, but there is no reason why the propellant choice could not be another design parameter if there is a database of propellant burning characteristics available. For ammonium perchlorate the erosive burning rate constant k is $1.0E{-4}$ s/ft, and the burning rate constants a and n are 0.15 and 0.4, respectively.

A nominal 50-lbf payload weight and 50-lbf electronics/actuators weight was also used, and it was assumed that the control system would not become active until 0.4 s after launch to give the dynamic pressure time to reach levels sufficient for vehicle control.

Link to GA

Linking the separate software codes to the GA was done in a modular fashion so that other modules could be later substituted for the ones used in this study. The GA passes all of the design variables down to the six-DOF via one subroutine call statement. The six-DOF then calls the other components, including the mass properties routine that determines component inertias and the center of gravity of the system.

Variables Governing Design

There are 9 variables that govern the SRM design, 14 variables that govern the external shape of the vehicle, 2 variables that control the launch angle (verticality and heading), 3 variables that define the autopilot performance, and 1 variable to set the effective navigation ratio or gain, for a total of 29 variables. Figure 1 shows the external geometry variables, with the exception that the nozzle exit radius, is actually determined from the expansion ratio (A_e/A^*), which is one of the SRM design variables. Though the nozzle is shown, it is merely for visualization purposes. The nozzle actually resides within the total length of the missile. The nozzle exit radius is not, however, free from external aerodynamic considerations because there is a substantial drag penalty that can be incurred if the nozzle exit radius exceeds the body radius. Ideally the GA will learn to design the rocket motor and external shape cooperatively so that good thrust levels are obtained without incurring a drag penalty. Basically, then, all outer body dimensions are controlled by the GA, from the nose length to the nozzle exit radius. Figures 2 and 3

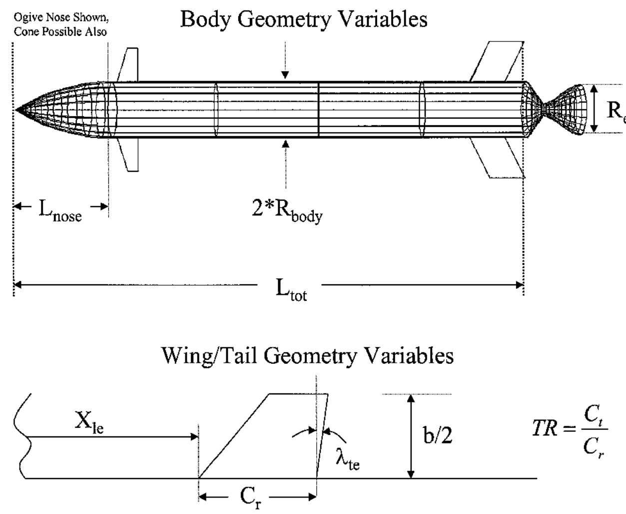

Fig. 1 External shape design variables.

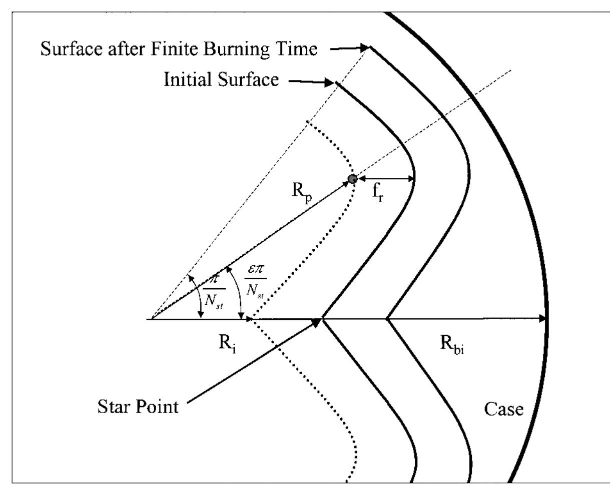

Fig. 2 SRM grain design variables.

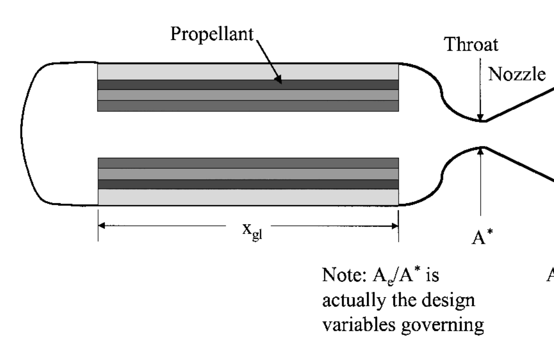

Fig. 3 SRM grain length, throat diameter, and expansion ratio.

show the nine SRM design variables that are also part of the design process, with (A_e/A^*) being one of those variables.

Table 1 formally shows the minimum, maximum, and resolution that is desired for each variable. The minimum, maximum, and resolution dictate the size of the optimization space. In GA terms, the number of genes in each chromosome (also known as the number of bits in a base-2 conventional binary encoding system of 1 and 0) is defined as

$$\text{number of genes} = \sum_{n=1}^{\substack{\text{number of}\\ \text{parameters}}} \left(\text{Integer}\left[LN\left(\frac{\max_n - \min_n}{\text{resolution}_n}\right) \Big/ LN(2) \right] + 1 \right) \tag{3}$$

Table 1 Maximum, minimum, and resolution of variable for guided system

Parameter	Minimum	Maximum	Resolution	Number of genes
R_{bi}	2.0	10.0	0.02	9
R_p	0.2	9.9	0.02	9
R_i	0.1	9.5	0.02	9
x_{gl}	50.0	200.0	1.0	8
N_{st}	3	10	1	3
f_r	0.05	1.0	0.01	7
ε	0.25	1.0	0.01	7
D^*	0.1	9.0	0.01	10
R_{exp}	1.0	20.0	0.2	7
Nose	0	1	1	1
L_{nose}	20.0	90.0	5.0	4
L_{tot}	50.0	450.0	10.0	6
R_{body}	3.0	20.0	1.0	5
b_w	0.0	80.0	1.0	7
C_{rw}	0.0	80.0	1.0	7
TR_w	0.0	1.0	0.1	4
λ_{tew}	0.0	44.0	2.0	5
X_{lew}	20.0	200.0	5.0	6
b_t	0.0	80.0	1.0	7
C_{rt}	0.0	80.0	1.0	7
TR_t	0.0	1.0	0.1	4
λ_{tet}	0.0	44.0	1.0	5
$\mathrm{X}_{\mathrm{let}}$	200.0	400.0	5.0	6
θ	5.0	90.0	1.0	7
ψ	0.0	180.0	1.0	8
ζ	0.2	1.0	0.05	4
τ	0.1	0.9	0.1	3
ω_{cr}	10.0	100.0	5.0	5
N'	2.0	5.0	0.1	5

The GA requires parameter bounds and resolutions only, and from Table 1 it is obvious that a very broad range of designs is possible. In fact, the specified bounds and resolutions, translated into the number of bits required to specify the parameters in binary code, means that 2^{175} possible designs exist.

Goals for Guided Interceptor

In order, the goals were the following: minimize miss distance, minimize intercept time, minimize takeoff weight, and minimize the maximum g loading experienced by the missile. Because an apportionment pareto GA was used, goal order is important because the ordering determines how many points are allocated to each goal.

Mode of Operation of GA

In this study, tournament selection with single-point crossover was chosen as the primary reproduction/mating scheme for the 150 population members chosen. The crossover rate was set to 90% and the mutation rate was 0.2%. Elitism was selected so that the best performer in each goal area was preserved intact from generation to generation. With the four goals listed, a domination strength of 4.0 was used, meaning that miss distance was four times more important that minimizing the maximum g loading.

Design Conflict Checking

Some obvious geometrical checks were used to keep the GA from expending computational resources for designs that were not practical. Seven separate checks were made, as follows: 1) Outer rocket motor case radius cannot exceed body radius. 2) Rocket motor grain length cannot exceed body length. 3) Tail control surfaces cannot be coincident with, or in front of, the wing. 4) Tail control surfaces cannot overhang the aft end of the missile. 5) Wing cannot overhand nose portion. 6) Based on the specified payload and electronic weights and densities, and the rocket motor size, the total volume of the missile must be able to house these components. 7) Tail control surfaces must be located such that the actuator hinge line (assumed to be at the 50% location of the tail root chord) can be placed at or very near the rocket motor throat. Because the actuators take up a considerable volume, it is logical that they would be placed near the throat of the nozzle. If any of these conflicts occur, the GA is sent back extremely poor performance values in each goal area so that it will learn not to try these designs in the future. Although this method is admittedly rigid and risks overlooking valuable information that might be gained from one discipline area even though another area fails, developing penalty functions among multiple disciplines can lead to situations where the researcher makes subjective judgements about the importance of different disciplines. For example, which solution is worse, one where the rocket motor explodes or one where the missile tumbles uncontrollably?

Table 2 Target definition

Variable, unit	Value
Initial downrange location, ft	120,000
Initial crossrange location, ft	50,000
Initial altitude, ft	250,000
Initial downrange velocity, ft/s	−3,000, toward interceptor
Initial crossrange velocity, ft/s	0
Initial vertical velocity, ft/s	4,500, down toward ground
Downrange acceleration, ft/s^2	30, decelerating toward interceptor
Crossrange acceleration, ft/s^2	0
Vertical acceleration, ft/s^2	−10, decelerating vertically toward interceptor

Thermal and Structural Considerations

The six-DOF software calculates stagnation temperature at each time step based on Mach number and altitude (standard atmosphere is assumed). If the stagnation temperature ever exceeds 2500°R, where many common missile construction materials begin to lose structural integrity, it is assumed that the missile fails either structurally or through reduced seeker performance.

Other structural considerations are manifested in the strength of the wings and tails because these are obvious weak points. Wing and tail loads during flight are used to calculate root bending moments and bending stresses. If the bending stresses ever exceed 185,000 psi (typical for stainless steel), the wing or tail surfaces fail, and the flight is terminated at that point. The wing joints are assumed to be rigidly connected to the missile body along the entire root chord. Each actuator-controlled tail surface is assumed to be mounted on a 1.25-in.-diam stainless steel rod. The GA must learn to design systems that will not fail either thermally or structurally. Designs that fail in either of these areas are allowed to retain the value of the miss distance at failure.

Target

The target specified for this research is a fast point-mass ground-attack reentry vehicle like a SCUD missile. For this study, Table 2 defines the target parameters.

Results

With the design problem and parameters completely defined, the apportioned pareto GA was executed for 50 generations. Because there were multiple goals involved in the design process, it is appropriate to examine the history of each goal area through the generations. Figure 4 shows miss distance and intercept time for several generations. It is clear that more members of the population maneuver closer to the target as the generations progress. In generation 14, 36 members of the population (36 out of 150) had a miss distance within 100 ft. By generation 50, 124 members of the population reached within 50 ft of the target. By inspection, it is also clear from Fig. 4 that intercept time falls appreciably between generations 30 and 50 even though the two populations have similar miss distance distributions. Of solutions with a 5-ft miss distance, intercept times varied between 38.8 and 42.3 s.

Takeoff weight and miss distance for the same four generations are shown in Fig. 5. It is obvious from Fig. 5 that the GA increased takeoff weight from generations 30 to 50 (likely through the addition of fuel) to yield the decreased intercept times shown in Fig. 4. This result was expected. However, note that continuing to increase fuel mass while decreasing intercept times has a limit when thermal considerations are involved. If the speed of the rocket becomes too high, the thermal loads will cause the rocket to fail, and the net result will be large miss distances, not decreased intercept times. Note that

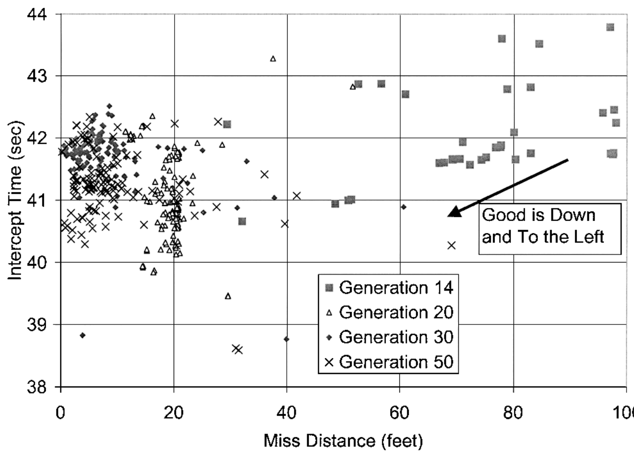

Fig. 4 Miss distance and intercept time: generations 14, 20, 30, and 50.

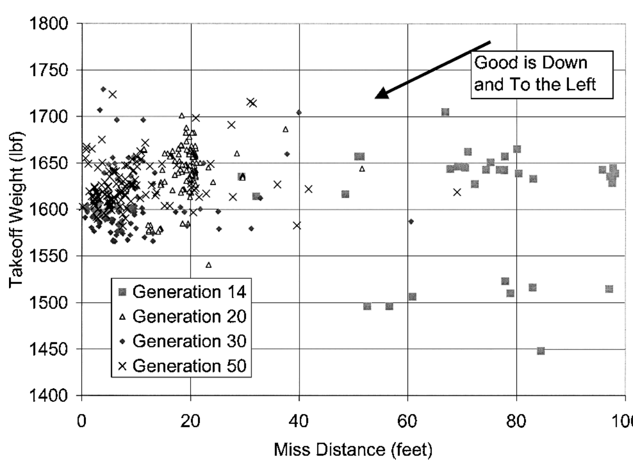

Fig. 5 Miss distance and takeoff weight: generations 14, 20, 30, and 50.

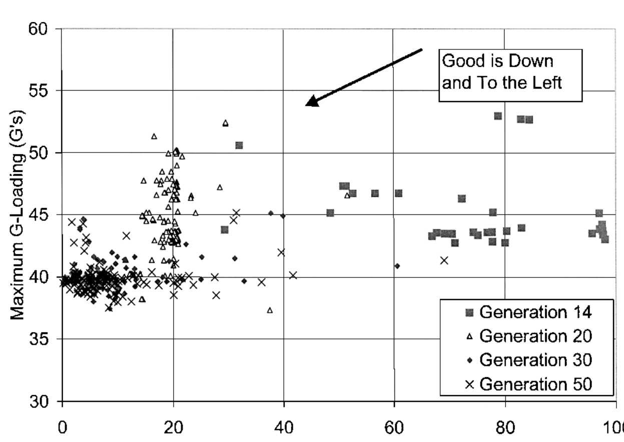

Fig. 6 Miss distance and *g* loading: generations 14, 20, 30, and 50.

the majority of the designs fall within 175 lb of a mean weight of roughly 1600 lb.

Figure 6 shows the maximum g loading and miss distance for these same four generations. Generations 30 and 50 show that about 40 g is the maximum g loading that can be expected during the missile flight. This g loading is certainly within the capability of existing electrical and mechanical systems. From generation 20 to generations 30 and 50, there is a clear trend toward minimizing the g loading. An average reduction of 5 g occurs between generation 20 and generation 50. There is no clear improvement between generations 30 and 50. As in any probability-based algorithm, there are

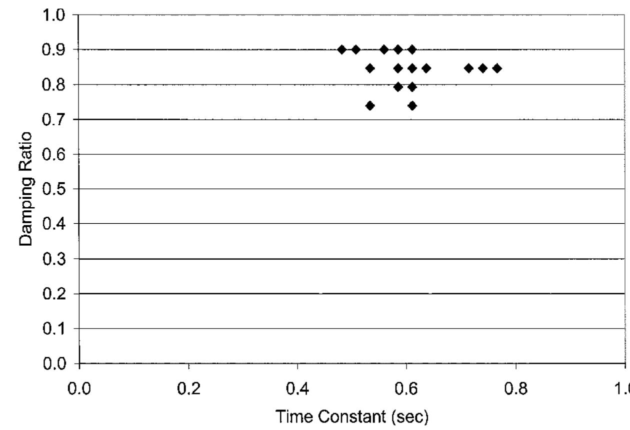

Fig. 7 Generation 50 time constants and damping ratios.

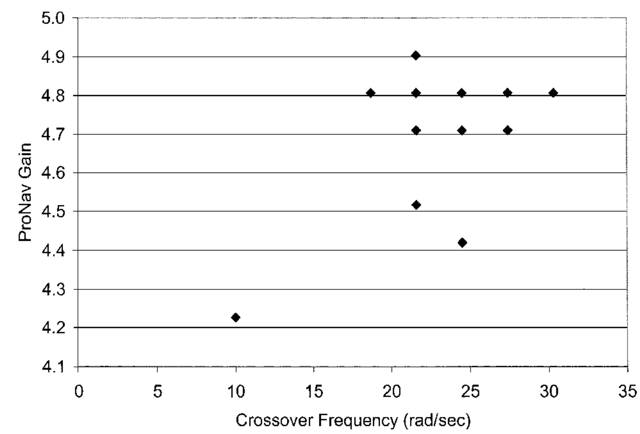

Fig. 8 Generation 50 proportional navigation gains and crossover frequencies.

no guarantees that continued execution cycles will result in better answers.

Figure 7 shows the prevalent time constants and damping ratios that dominated generation 50. There are clearly not 150 individual points (representing members of the population) in Fig. 7 because by generation 50 many members of the population were using exactly the same damping ratios and time constants. High system damping is obviously preferred. This result should be expected because overshooting acceleration commands is not a desirable missile flight characteristic because it wastes energy. The preferred time constants were in the 0.5–0.6 s range, which is very reasonable because the target is not conducting evasive maneuvers to escape the interceptor. Missiles that are designed to intercept high-g (9 g is typical) maneuvering targets have time constants near from 0.2 to 0.3 s so that they can quickly respond to target evasive tactics.

Figure 8 shows the proportional navigation gains and crossover frequencies that dominated the population at generation 50. Fairly high navigation gains (4.7–5.0) dominated the population, which means that the system quickly tried to minimize heading errors. Low values of the navigation gain, in the 2.0–3.5 range, would tend to delay correcting heading errors. For high-altitude intercept missions, it makes sense to take out heading errors early in the flight rather than waiting until the altitude is such that system responsiveness suffers from the lack of air density, that is, dynamic pressure. The dominant crossover frequencies were between 20 and 30 rad/s. This result is not surprising because the highest value that could safely be used[15] was roughly 41.66 rad/s, which corresponds to one-third of the bandwidth of the actuator used in this study.

Before proceeding to look at some of the designs developed by the GA, note which initial Euler launch angles were preferred (see Fig. 9) at generation 50. The launch angles were in a fairly tight band, generally between 52 and 56 deg verticality with a heading from 60 to 63 deg. None of the trajectories flown by the three solutions down

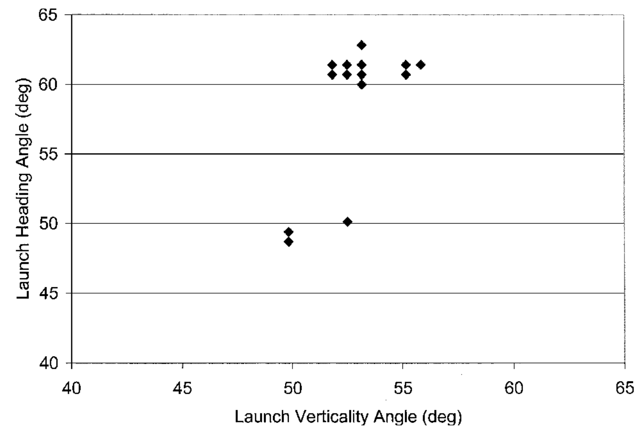

Fig. 9 Generation 50 preferred Euler launch angles.

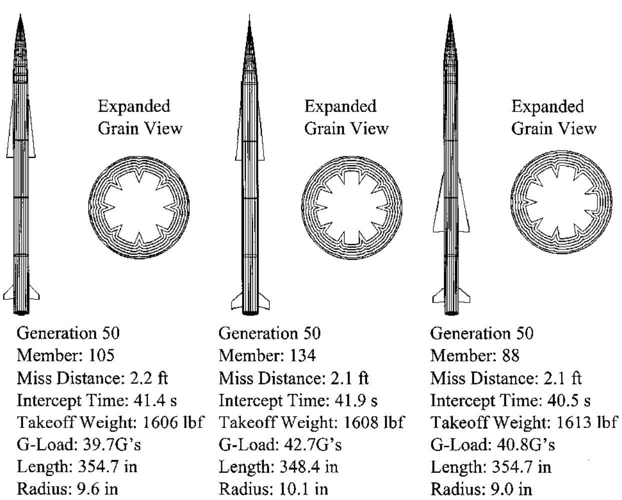

Fig. 10 Three generation 50 designs with miss distances less than 3 ft.

near the 50-deg heading came within 20 ft of the target, and so it is clear that the best launch angles were those with a near 60-deg initial heading. The target initially is located at a 67.38-deg heading angle, which decreases with time because of the negative downrange velocity component; thus, the GA has chosen to lead the target by a few degrees at launch. A 7-deg lead angle at 300,000 ft of slant range translates into a lead distance of 36,560 ft. The target also has an initial elevation angle of 62.52 deg (decreasing also because it is descending), and the GA has chosen to lead the target vertically by a similar magnitude at launch. Even though the initial interceptor velocity is zero, and proportional navigation might tend to make the system believe that very large lead angles are required, the GA has learned not to overcompensate the launch lead angles.

Although neither the launch angles nor the autopilot performance parameters show great variation among the members of the population, the actual missile designs produced during the solution process are quite diverse. The designs shown in Figs. 10 and 11 all come from generation 50, the final generation, and are segmented into designs that have miss distances less than 3 and 1 ft, respectively. Not all of the designs that meet these accuracy criteria are shown, these particular designs are merely representative of some of the more accurate designs in the population. By generation 50, the GA has found two basic classes of external designs that work fairly well: highly tapered, small semispan, forward-placed wings and moderately tapered, moderate semispan, aft-placed wings. The GA found that large semispan designs produce large bending moments (which tend to cause structural failure), and so wing areas were held at reasonable levels by increasing the root chord. The tail surfaces are fairly similar, though the placement of the surfaces varies slightly as does their size. The placement of the control surfaces dictates where the throat of the rocket motor is placed (to make room for the actuators within the missile body), and so the length of the nozzle expansion region varies. In no case, however, did the actual exit area

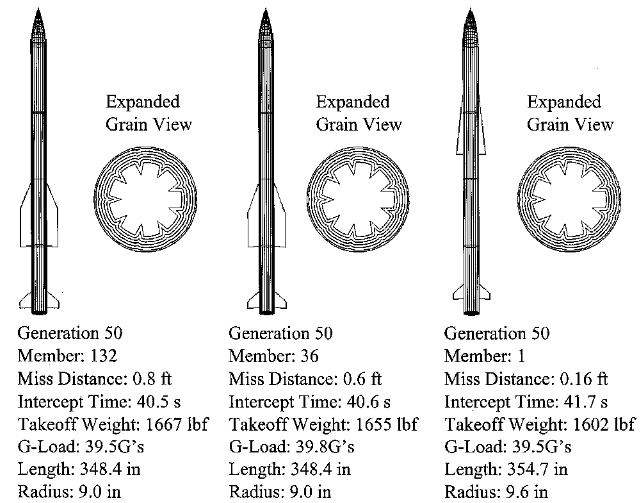

Fig. 11 Three generation 50 designs with miss distances less than 1 ft.

exceed the diameter of the missile body. Designs yielding nozzle exit radii exceeding the radius of the missile body would produce excess drag, and the GA learned to avoid these types of designs. The rocket motor grain designs appear to be very similar, but Fig. 10 shows examples of 8-, 9-, and 10-pointed star grains. All of the designs shared a large initial burning area and large combustion chamber.

Nose shapes also continue to vary between ogives and cones, although it appears that the ogive nose exists in the more accurate examples. Physical sizes and takeoff weights of the interceptors are very similar, as are intercept times and maximum g loads. The three best designs (in terms of miss distance only) all had nine-pointed star grains, but the wing sizes and locations still vary significantly. Also note that these designs all had nose shapes that were fairly blunt compared to the other designs that were slightly less accurate. These nose shapes were not blunt enough to incur a drag coefficient penalty larger than 0.012 based on a Sears–Haack body, and so an examination of the aerodynamic data for these shapes revealed that the net effect of the change in the nose length was to move the center of pressure farther forward very slightly (an average of approximately 1.5 in.) over all flight conditions. This center of pressure movement helped reduce the static margin at rocket motor burnout, thereby increasing maneuverability at the coast condition without seriously impacting maneuverability during rocket motor burn. Simply moving the wings slightly forward might have had the same net effect at burnout, but this wing movement would have certainly changed the maneuverability more during rocket motor burn when initial acceleration commands are more rigorous (to get the interceptor on the correct course).

The 50 generations presented here required 10 days of CPU time on a Silicon Graphics R-10000 processor.

Conclusions

The apportioned pareto GA is well suited to designing complete interceptor systems consisting of propulsion, aerodynamics, and autopilot modules. This work has shown that an all-at-once design process, controlled by the proper intelligent system tool, can provide many candidate interceptors capable of defeating a fast-moving target. Unlike gradient-based optimization approaches, GAs find good designs by learning their own design lessons, without having to be given a starting solution or sensitivity derivatives. This work also shows that GAs can easily handle continuous and discrete variables and work very well when given system-level performance goals rather than discipline-specific goals.

Simple constraints, such as the thermal limits and structural integrity calculations used in this work, provide a means of injecting real-world considerations into the design process. Even with diverse performance modules and diverse goals, the GA was able to learn how to design around the constraints while achieving good performance in overall system goals. In this difficult high-altitude/high-speed engagement scenario, the GA developed multiple designs capable of close intercept.

References

[1]Holland, J., *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA, 1975, pp. 1–207.

[2]Louis, S. J., and Rawlins, G. J. E., "Pareto Optimality, GA—Easiness and Deception," *Proceedings of the Fifth International Conference on Genetic Algorithms*, Morgan Kaufmann, San Mateo, CA, 1993, pp. 118–123.

[3]Von Neumann, J., *Theory of Self-Reproducing Automata*, completed and edited by A. W. Burks, Univ. of Illinois Press, Urbana, IL, 1966, pp. 1–53.

[4]Farmer, D., Lapedes, A., Packard, N., and Wendroff, B., *Evolution, Games, and Learning*, North-Holland, Amsterdam, 1986, pp. 187–204; reprint *Physica D*, Vol. 22D, Nos. 1–3, 1986.

[5]Langton, C. G., *Artificial Life: An Overview*, MIT Press, Cambridge, MA, 1995, p. 340.

[6]Norris, S. R., and Crossley, W. A., "Pareto-Optimal Controller Gains Generated by a Generic Algorithm," AIAA Paper 98-1010, Jan. 1998.

[7]McGookin, E. W., Murray-Smith, D. J., Li, Y., and Fossen, T. I., "Parameter Optimisation of a Non-Linear Tanker Control System Using Genetic Algorithms," *Proceedings of the 2nd IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems*, IEEE Publications, Piscataway, NJ, 1997, pp. 37–42.

[8]Martin, R. C., IV, "A Gain Scheduling Optimization Method Using Genetic Algorithms," M.S. Thesis, Air Force Inst. of Technology, AD-A289306, Wright–Patterson AFB, OH, Dec. 1994.

[9]Karr, C. L., and Harper, T. R., "Genetic Algorithms in Adaptive Fuzzy Control," *Conference Proceedings from the North American Fuzzy Logic Processing Society Meeting (NAFIPS 1992)*, Vol. 2, North American Fuzzy Information Processing Society, Lakeland, FL, 1992, pp. 506–514.

[10]Perhinschi, M. G., "A Modified Genetic Algorithm for the Design of Autonomous Helicopter Control System," AIAA Paper 97-3630, Aug. 1997.

[11]Karr, C. L., Freeman, L. M., and Meredith, D. L., "Genetic Algorithm Based Fuzzy Control of Spacecraft Autonomous Rendezvous," *Proceeding of the Fifth Conference on Artificial Intelligence for Space Applications*, NASA Marshall Space Flight Center, Huntsville, AL, 1990, pp. 43–51.

[12]Homaifar, A., and McCormick, V. E., "Full Design of Fuzzy Controllers Using Genetic Algorithms," NASA Center for Aerospace Research at North Carolina A&T State Univ., Contract NASA-N93-194520 6-80, Dec. 1992.

[13]Clergen, J. B., "Solid Rocket Motor Conceptual Design, the Development of a Design Optimization Expert System with a Hypertext User Interface," AIAA Paper 93-2318, June 1993.

[14]McCain, J.W., "A Variable Selection Heuristic for Solid Rocket Motor Design," Ph.D. Dissertation, Dept. of Mech/Aero Engineering, Univ. of Alabama, Huntsville, AL, Jan. 1993, Univ. Microfilm DA9324488.

[15]Zarchan, P., *Tactical and Strategic Missile Guidance*, 3rd ed., Vol. 176, Progress in Astronautics and Aeronautics, AIAA, Reston, VA, 1997, pp. 461–584.

[16]Washington, W. D., "Missile Aerodynamic Design Program," Army Aviation and Missile Command, Rept. 35898, Redstone Arsenal, AL, 1990.

[17]Ashley, H., and Landahl, M., *Aerodynamics of Wings and Bodies*, 1st ed., Dover, New York, 1965, pp. 89, 90.

[18]Burkhalter, J. E., "Rocket Propulsion II Course Notes," Aerospace Engineering Dept., Auburn Univ., Auburn, AL, 1985.

[19]Sforzini, R. H., "Notes on Solid-Propellant Rocket Motors Including Jet Propulsion Fundamentals," Auburn Univ. Engineering Extension Service, Auburn, AL, 1983.

[20]Etkin, B., *Dynamics of Atmospheric Flight*, 1st ed., Wiley, New York, 1972, pp. 60–123.

D. B. Spencer
Associate Editor