JOURNAL OF SPACECRAFT AND ROCKETS
Vol. 46, No. 1, January–February 2009

Genetic-Algorithm Optimization of Liquid-Propellant Missile Systems

David B. Riddle,* Roy J. Hartfield,† John E. Burkhalter,‡ and Rhonald M. Jenkins§
Auburn University, Auburn Alabama 36849

DOI: 10.2514/1.30891

The use of computer programs driven by genetic algorithms has become an increasingly popular method for optimizing engineering designs. This paper focuses on the modeling and optimization of liquid-rocket-engine-propelled missiles with a computer program that is composed of a series of codes that simulate the performance of liquid-propelled rockets and are controlled by a genetic algorithm. Because the entire missile design is considered, the complete system performance must be modeled accurately and efficiently. The methodology described in this paper has been extended to both the preliminary design and reverse-engineering of liquid-propelled missiles. The performance model has been validated against the performance of a liquid-propelled missile and the results are provided. Two different fast-aerodynamic-prediction codes are used and their results are compared. A complete preliminary design of a liquid-propelled missile system is considered with a variety of goals and constraints. Results from the preliminary design optimization are also shown and discussed in detail.

Nomenclature

f = fuel-to-oxidizer ratio
g = acceleration due to gravity
h_e = exit enthalpy
h_0 = total enthalpy
I_{sp} = specific impulse
M = Mach number
q = dynamic pressure
u_e = exit velocity
ϕ = equivalence ratio

I. Introduction

LIQUID-PROPELLED missiles are complex systems with a broad range of design options. Predicting the performance of a missile for which only a limited amount of information is available is challenging and is not always possible. Genetic algorithms (GAs) have proven to be very effective at optimizing a variety of engineering designs and are now being used to reverse-engineer or predict the capabilities of missile systems with only a few known design variables. Although it is unlikely that a GA would converge to the exact missile design of interest, GAs are able to efficiently produce multiple close matches to known solutions [1–3].

Pioneered by Holland [4] in the 1970s, GAs are based on the biological concept that a species evolves or adapts over successive generations. Traits that improve fitness or performance are passed from one generation to the next and unfavorable characteristics are gradually eliminated. Over the course of many generations, the average fitness of the members in a population will gradually begin to increase and the design will converge toward the objective. Depending on the setup, there can be multiple goals and the GA attempts to find the combination of characteristics that results in a performance that matches these goals [5].

Genetic algorithms can be used to optimize virtually any system with multiple design variables for which the performance can be computationally simulated. The use of genetic algorithms has become increasingly popular in the optimization of engineering designs and has already been used extensively in the aerospace industry to forward-optimize designs, in which constraints on performance drive the design process. The design of helicopters [6], spacecraft controls [7], flight trajectories [8], gas turbines [9], airfoils [10], boosted ramjets [11], interceptor missiles [12,13], aircraft [14], hybrid rockets [15], liquid-propelled rockets [1,16–18], and solid rockets [19,20] have all benefited from the application of GAs in this manner.

Genetic algorithms have also been shown to be useful in reverse-engineering problems: specifically, in reverse-engineering missile designs. In contrast to design optimization, reverse-engineering assumes that some of the performance goals or design variables are already known, and the GA attempts to discover a variable set that matches the known goals. With a limited set of initial data, it was determined that a GA is better able to establish the remaining unknown design parameters than a trial-and-error method [1]. In a precursor to the current research, Burkhalter et al. [1] developed a GA-based program that was able to partially reverse-engineer ballistic solid rocket missiles. Although their program was able to discover much of the external design and some performance characteristics, some of the design variables could not be accurately identified. Using a derivative of this program, Metts [5] conducted much more extensive research on the performance of the GA in reverse-engineering ballistic missile designs. He found that the GA was capable of finding quality reverse-engineering matches, but human inspection was still necessary to determine the true best performer from a small group of optimized missile designs. This scenario often occurs because of the fact that multiple designs can achieve identical performance goals.

Building directly on the work done by Burkhalter et al. [1], the missile optimization code has been modified to use the GA in optimizing liquid-fueled ballistic and aerodynamically controlled missiles. The optimization code is a combination of many individual codes that together simulate the performance of a missile and determine its fitness in relation to the other candidate designs. These codes predict performance by analyzing a candidate missile design's aerodynamics, mass properties, propulsion characteristics, and guidance and control system. The 6-degree-of-freedom (6-DOF) model ties everything together, providing the performance data for each missile design. The GA then determines and operates on the best

Presented as Paper 362 at the 45th AIAA Aerospace Sciences Meeting and Exhibit, Reno, NV, 8–11 January 2007; received 8 March 2007; revision received 19 November 2007; accepted for publication 3 December 2007.

*Graduate Student, Aerospace Engineering; currently Aerospace Engineer, Gray Research, Huntsville, AL. Member AIAA.
†Professor, Aerospace Engineering. Senior Member AIAA.
‡Professor Emeritus, Aerospace Engineering. Associate Fellow AIAA.
§Professor Emeritus, Aerospace Engineering, Senior Member AIAA.

solutions for a given generation of designs, depending on the user-defined objective function. Additional features of the current tool include the development of a version of the code that employs an alternate aerodynamic analysis code known as Missile Datcom [21] and variable specific impulse I_{sp}. These improvements will be discussed in detail, along with a brief summary of each of the major performance codes.

II. Background and Theoretical Development

The liquid-propelled rocket optimization code is composed of many individual codes that work together to predict the performance of each missile design selected by the GA for analysis. The major codes will be discussed briefly, with special attention paid to those that have been upgraded. Some background information on the GA will first be provided, followed by discussion of the aerodynamics, mass properties, 6 DOF, liquid propulsion system, and guidance system models.

A. Genetic Algorithm

The GA employed in the effort uses binary encoding of the design variables to define a particular missile system. The optimization process begins with the user selecting a range and resolution for each of the design variables. By randomly selecting values for each design variable from within the user-defined boundaries, the GA then generates a population of candidate designs. Each candidate design is analyzed by the suite of performance codes. The number of members in a generation and the number of generations that the GA should run are also values set by the user. Next, the GA ranks the members according to the performance of each as determined by the objective function. The GA employed in this effort uses a tournament selection process that chooses the best-performing member from a randomly selected pair. The resulting member is then mated with the best performer from a different pair. The mating process combines the two members' genetic material, which is stored in binary form. The subsequent generation is filled with members that result from the mating process. The process is repeated as each member in the new population is analyzed by the performance codes and its fitness is ranked.

Over successive generations, the fitness of the designs will improve as desirable designs are passed on, and undesirable designs are eliminated. The speed and efficiency of this process is highly dependent on the ranges of the GA variables and the number and types of goals. Allowing wide variations in each of the design variables greatly increases the number of possible solutions that the GA must analyze. However, if several design variables are already known, as is usually the case in reverse-engineering problems, a solution can be reached much more quickly [5].

The objective function has an arguably greater effect on the efficiency of the optimization code. Experience has shown that given one goal, the GA usually finds a solution using a relatively small number of calculations of the objective function. As more goals are added, however, optimal solutions can be more difficult to find, as the goals are often competing with one another. For example, the user might wish to maximize range and minimize the system weight of a missile design. Those goals are in direct competition, and finding the balance of range and weight based on the goal description is computationally expensive. In addition, the user must keep the goals evenly proportioned while adjusting a weighting factor to emphasize the more important goal if necessary. Many of the settings and options that control the behavior of the GA are found in the GA input file [2,5,11].

The 27 GA variables that are considered in the optimization program are as follows: propellant type, equivalence ratio, maximum chamber pressure (psi), nozzle throat area (in.^{2}), nozzle expansion ratio, fractional nozzle length, burn time (s), payload mass (lb), missile body diameter (ft), nose length (lnose/dbody), nose diameter (dnose/dbody), finset 1 root chord (cr/dbody), finset 1 taper ratio, finset 1 leading-edge angle (deg), finset 1 semispan (b2/dbody), location of the leading edge of finset 1 (% total length), finset 2 root chord (cr/dbody), finset 2 taper ratio, finset 2 leading-edge angle (deg), finset 2 semispan (b2/dbody), location of the trailing edge of finset 2 (% total length), autopilot time on delay (s), autopilot time constant (s), autopilot damping coefficient, crossover frequency, gain setting for the proportional navigation algorithm, and initial launch angle (deg).

Clearly, all of the possible parameters affecting the design of liquid powered missiles could not be encapsulated in just 27 variables. These variables were selected as being the most critical design options and have the greatest effect on a missile's performance. The remaining design options, which are not variable, are defined elsewhere including: constants, material densities, component masses, target and launch data, external geometry variables, reverse-engineering data and many other critical pieces of information. Detailed design parameters such as turbopump internal parameters, fasteners, and detailed plumbing and wiring schematics are not considered in this process. For the preliminary design variables used in the numerical construction of the missiles in this effort, the ranges and resolution are chosen to based on either design constraints, known parameters or both and member length.

B. Aerodynamics

Aerodsn [22] is a fast-aerodynamic-prediction analysis code that has been used successfully in several different versions of the GA missile optimization program. Developed by the U.S. Army Missile Command in the 1980s, it is a robust code restricted to axisymmetric cruciform missile shapes. More accurate computational fluid dynamics (CFD) codes are available, but the high computational cost makes CFD impractical for use with the GA in the preliminary design mode. Aerodsn uses the vehicle geometry and other parameters necessary for successfully generating an aerodynamic database. The required aerodynamic data are generated for the complete Mach number range and a complete missile orientation sweep for the 6-DOF to determine the aerodynamic loads and moments under any flight condition. The missile is assumed to be symmetric, and so the yawing moments are determined from the pitching-moment coefficients, and the side forces are determined from the normal force coefficients.

Although Aerodsn has been used successfully as the aerodynamic prediction code for the GA program, it was determined that the program would benefit from a full-featured and industry-standard aerodynamic code. Consequently, an alternate version of the optimization suite has been developed that uses Missile Datcom. Missile Datcom [21] is also a fast-aerodynamic-prediction missile analysis tool with a wide range of capabilities.

Coding issues have made it impractical to implement Missile Datcom in the final version of the liquid-propelled-missile GA program, but it can be used to corroborate results obtained from the Aerodsn version. An extensive study was made comparing the two aerodynamic codes and extensive results have been obtained. The trajectory plot, presented later, illustrates how similar the predictions of two aerodynamic codes were to each other. The close agreement in the maximum range, as well as the flight path, suggests that Aerodsn and Missile Datcom are providing comparable results for the aerodynamic coefficients. A complete set of results is presented in a later section of this paper.

C. Mass Properties

The mass-properties routine is a comprehensive analysis of the missile system's component masses and moments of inertia about all three axes. Cross-product moments of inertia are assumed to be zero because missile symmetry is assumed. The moment of inertia for each of the missile components is calculated. Once the center of gravity (c.g.) of the missile is determined, the moments are transferred to the c.g. and subsequently stored for future use. In the same routine, the mass and moment of inertia of the fuel and oxidizer are also determined as a function of time. This information, with the time-dependent thrust data, is written to a file (TMASS.DAT) that is later used by the 6-DOF model. The components that are considered in the analysis are shown in Table 1. Also listed in the table are six

Table 1 Mass-properties components

Part no.	Component
1	Box 1
2	Avionics or electronics
3	Compressed gas for pressurization
4	Compressed gas tank
5	Fuel
6	Fuel tank
7	Oxidizer
8	Oxidizer tank
9	Engine assembly
10	Nozzle
11	Nose-cone fairing
12	Cylindrical (main) fairing & wiring
13	Aft fins
14	Gimbals
15	Warhead
16	Forward fins
17	Servos
18	Box 2
19	Box 3
20	Box 4
21	Box 5
22	Box 6

generic boxes that have been built into the code. These boxes allow the user to place custom payloads or other equipment inside the missile. The boxes can also be positioned at any location within the missile, with the exception of box 1, which is always located at the nose.

D. Six-Degree-of-Freedom Model

The 6-DOF routine is based on the equations of motion found in [23] and uses an Earth-centered coordinate system. It is assumed that the missile is rigid, that all masses are stationary, and that all cross products of inertia are negligible. The equations of motion accept aerodynamic data from Aerodsn or Missile Datcom, mass and moments of inertia from the mass-properties routine, and thrust data from the liquid-propelled rocket module, and other required information is passed through the array. The 6-DOF routine uses a 7th–8th-order Runge–Kutta numerical integration routine to simulate the flight of the missile. The time step is a variable and is dependent on the magnitude of the largest derivative in the equations of motion [13]. The flight of the missile is recorded, and necessary information is stored in an array and passed back to the GA routine for further analysis.

E. Liquid Propulsion System

The liquid-propelled rocket model is based on a generic set of required parameters that define the operation of a single-stage liquid-propelled rocket engine. The oxidizer and fuel are stored in cylindrical tanks with hemispherical end caps that are sized based on the specific missile design. Standard equations are used to predict the thrust for a variety of fuel and oxidizer combinations at different operating combustion pressures and throat areas. Variation of the thrust due to altitude change is also accounted for in the equations. The pressure and temperature in the combustion chamber are assumed to be constant, and it is also assumed that the thrust instantly drops to zero at burnout. The nozzle is designed to connect to the combustion chamber and can be configured to extend aft of the base of the missile or to have its exit flush with the base of the missile. The system assumes that turbopumps are required to compress the fuel and oxidizer before injection into the combustion chamber, and the fuel and oxidizer tanks are under relatively low pressure (around 100 psia). All necessary plumbing and wiring are included as uniformly distributed mass [20].

The propellant is a design variable that can be chosen from a list of 29 included fuel and oxidizer combinations. Propellant properties such as stoichiometric mixture ratio, combustor total temperature, molecular weight, characteristic exhaust velocity, sea-level specific impulse, and ratio of specific heats are included in the list with their corresponding fuel type. As a result of a recent upgrade, propellants with sufficient data have been approximated with a curve fit to determine specific impulse as a function of equivalence ratio.

Table 2 Variable I_{sp} propellant combinations

Propellant name[a]	I_{sp} (s) at $\phi = 1$
IRFNA/UDMH	274
IRFNA/hydrazine	287
IRFNA/RP-1	257
IRFNA/JP-4	254
IRFNA/MMH	268
N_2O_4/UDMH	284
N_2O_4/kerosene	283
N_2O_4/hydrazine	291
N_2O_4/MMH	301
LOX/hydrazine	314
LOX/UDMH	309
LOX/LH_2	367

[a]IRFNA is inhibited red fuming nitric acid, UDMH is unsymmetrical dimethylhydrazine, RP is refined petroleum, JP is jet propellant, MMH is monomethylhydrazine, LOX is liquid oxygen, and LH_2 is liquid hydrogen

A method for determining the specific impulse I_{sp} as a function of the equivalence ratio ϕ has been implemented in the optimization code. This allows the I_{sp} to be calculated based on the equivalence ratio that the GA chooses. It should be noted, however, that some of the propellants available in the program are rare, and the necessary thermochemical data for these propellants were not readily available. As a result, these propellant combinations are still included but have a constant specific impulse. The fuels that currently have variable I_{sp} capability are listed in Table 2.

The process of characterizing the thermodynamic conditions in a combustion chamber is highly complex. The large amounts of individual chemical species that appear make solving the chemical equations by hand impractical. Computer codes are routinely used to determine these values instead. STANJAN [24] is one such code and was used in calculating the I_{sp} as a function of equivalence ratio for the propellants in the optimization program. STANJAN solves for the thermodynamic conditions using linear programming to minimize the Gibbs free energy.

The specific impulse values for the propellants with sufficient data are approximated by a fifth-order curve fit that can be used to obtain the I_{sp}, given an equivalence ratio. The equivalence ratio is defined as

$$\phi \equiv \frac{f}{f_{\mathrm{stoich}}} \tag{1}$$

where f is the fuel-to-oxidizer ratio by mass [25]. The I_{sp} curve is valid for equivalence ratios between 0.25 and 4.0 for most of the propellant combinations. STANJAN was used to determine the I_{sp} at approximately eight equivalence ratios for each fuel and oxidizer combination. Given the initial enthalpy of the propellant, STANJAN calculates the total temperature in the combustion chamber, assuming 1000 psi (68 atm) total pressure. Assuming that entropy is constant, STANJAN then calculates the exit enthalpy of the flow when the nozzle exit pressure is 14.7 psi (1 atm). The exit velocity is calculated by hand and then divided by the acceleration due to gravity to obtain the specific impulse, as shown in Eq. (2):

$$I_{\mathrm{sp}} = \frac{\sqrt{2(h_0 - h_e)}}{g} = \frac{u_e}{g} \tag{2}$$

The resulting value is a reference I_{sp}, and the actual performance is adjusted based on the flow characteristics and the selected chamber pressure. This adjustment is done using a thrust coefficient and the nozzle expansion ratio. The chamber pressure directly impacts the thrust, and the exit velocity is adjusted based on the nozzle exit pressure. The nozzle exit pressure is calculated directly and a direct substitution is made into the thrust coefficient. The nozzle expansion

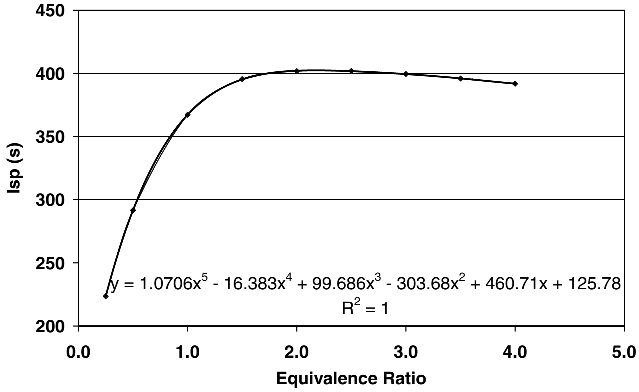

Fig. 1 Specific impulse as a function of equivalence ratio.

ratio is a GA design variable, and the nozzle exit area does not necessarily match the missile body diameter; however, the nozzle exit diameter is constrained to not exceed the body diameter. The mixture ratio is also adjusted based on the chosen equivalence ratio so that the masses of the oxidizer and fuel, as well as the size their respective tanks, are calculated correctly.

Figure 1 is a plot of the I_{sp} as a function of equivalence ratio for one of the available propellant combinations (LOX/LH_2). A trend line is fitted to the curve, and the corresponding fifth-order equation is also shown on the plot. Similar plots were made for each propellant combination, and the equation of the trend line corresponding to each propellant type was inserted into the liquid fuel's subroutine. The curve fits were compared with published data [26] to validate the results. Except for propellant combinations containing hydrogen, the maximum specific impulse occurs very near the stoichiometric fuel-to-oxidizer ratio. The I_{sp} is higher for the fuel-rich LOX/H_2 mixture, because the molecular weight of the fuel (H_2) is much lower than the molecular weight for the oxidizer (O_2).

F. Guidance System and Autopilot

The guidance system is based on the proportional navigation guidance law. The system attempts to rotate the missile at a rate proportional to the rate at which the line of sight to the target is moving. The guidance system is a two-axis feedback-control system that uses the pitch and yaw acceleration rates, but does not factor in the roll rate. The autopilot generates the elevator and rudder commands based on the acceleration rates determined in the guidance routine. The GA variables used by the guidance system and autopilot are the time delay, autopilot time constant, damping coefficient, crossover frequency, and the gain setting for the proportional navigation algorithm.

III. Results

The liquid-propelled missile system model has been validated against a known missile configuration and the results are provided. Performance data for a generic short-range ballistic missile similar to the foreign missile designated as a SCUD-B are given. Results from a simulated flight of the same missile design are compared with the known performance data. Results from unguided-missile optimization runs with either Aerodsn or Missile Datcom producing the aerodynamic coefficients will also be presented. The next section of results examines missiles produced by Aerodsn and Missile Datcom with aerodynamic control systems. Simulating aerodynamically controlled designs highlights the differences between the aerodynamic codes and offers a much more complete comparison than the unguided cases.

A. Model Validation

The liquid-propelled missile performance model was validated by flying a single-run case with a configuration very similar to the SCUD-B. A diagram of the SCUD-B is shown in Fig. 2, and the corresponding missile data [27] is given in Table 3. This set of data was input into the performance model along with additional data such as the fin geometry, and the resulting design was flown by the 6-DOF simulation. The I_{sp} and other fuel parameters were set manually in the liquid fuel's subroutine, and both Aerodsn and Missile Datcom were used as aerodynamic predictors for this case. One difference of note is that the SCUD-B has a conical nose section, whereas the simulated missile design has a blunted ogive nose. This fact, however, should have little impact on the performance of the simulated missile when compared with the SCUD-B.

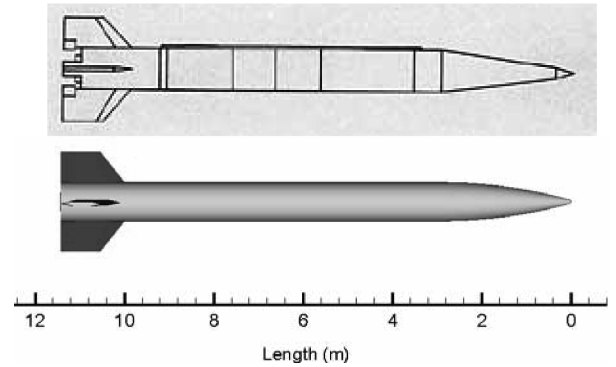

Fig. 2 Missile comparison.

The results of the model validation are given in Table 4, and a rendering of the resulting missile compared with the SCUD diagram is shown in Fig. 3. All other design parameters not listed in Table 3 were either direct inputs into the model or no data were available for comparison. The results show close agreement between the known missile's performance and the performance of the model that used Aerodsn. Missile Datcom predicts the range to be about 25 km farther than the actual SCUD. For both models, the major design parameters are only a few percent off from the actual values. This result provides a high level of confidence that the missile performance model using Aerodsn produces accurate results.

B. Unguided Results

The following results are from a series of unguided-missile optimization runs. The primary objective of this set of missile optimizations was to obtain missile designs using both aerodynamic prediction routines so that the results could be compared. The optimization codes are identical aside from changes made to accommodate their respective aerodynamic prediction codes. The goal of each optimization run was to determine the viability of a single-stage liquid-propelled ballistic missile with specific constraints, delivering a payload to a range of 700 km. The primary

Table 3 SCUD-B published data

Parameter	Value
Missile length	10.9 m
Missile diameter	0.88 m
Warhead weight	1000 kg
Thrust	13.35 t
I_{sp}	238 s
Burn time	62 s
Chamber pressure	6.71 MPa
Nozzle expansion ratio	10.32

Table 4 SCUD-B data comparison

Parameter	Known	Model
Missile length	10.93 m	11.40 m
Range	299.4 km	304.1 km
Launch weight	5828 kg	5769 kg
Thrust	13.35 t	13.18 t

Table 5 Case setup

Case	Aerodynamics	Mach limit
1	Aerodsn	7.1
2	Aerodsn	8.5
3	Datcom	7.1
4	Datcom	8.5

constraints included a maximum thrust limit of 28 t, IRFNA/RP-1 propellant, and a payload of 2000 kg.

To determine the feasibility of such a missile, the GA was configured to run a single objective-function-goal case to match a range of 700 km. The GA could have alternatively been set up to maximize range, but either strategy would determine whether a missile with these requirements could achieve the range goal. Matching the thrust and payload could have been additional GA goals, but including these goals would have complicated the runs. The optimization program works more efficiently when only a single goal is employed, and so limiting the number of goals to only those that are high priorities is advantageous. To account for these limitations though, a thrust ceiling was fixed in the program, and the payload, being one of the 27 GA variables, was limited to be 2000 kg in all cases. Table 5 shows the major differences between each run. The aerodynamics column identifies which code was used for the aerodynamic prediction for a particular case and the Mach limit column lists the maximum allowable Mach number for each case. All of the cases were run to 100 generations with 300 members in each generation. The convergence history of each case was recorded and all of the cases were found to be sufficiently converged with this configuration.

A note should be made about the Mach limit, which has not been mentioned until now and is listed in Table 5. Aerodsn uses supersonic theory to predict the aerodynamics, and the accuracy of this method becomes marginal in the hypersonic range. However, an extrapolation of the theory in this range is adequate for preliminary designs such as those in question. To account for this fact, an upper bound on the Mach number was fixed at 7.1 in Aerodsn. As the plots show, the optimized missile designs were not able to achieve the goal range with this configuration. Close examination revealed that the Mach number bound was a limiting factor in the missile performance. Revisions made to the Aerodsn code before this study provided an increased level of confidence in the aerodynamic predictions at high Mach numbers. The Mach number limit was raised accordingly until the GA was able to find a design capable of reaching the target. The new Mach number upper bound of 8.5 was used in conjunction with the previous two Mach 7.1 optimization runs to further support the comparisons of Aerodsn and Missile Datcom.

Performance plots generated from these optimization runs are provided in Figs. 3–7. Figure 3 plots the trajectory of each of the four resulting missile designs. The Aerodsn Mach 7.1 case flies slightly higher and farther than did the comparable Missile Datcom design, but the two Mach 8.5 missiles have almost identical trajectories. It is important to note that significant differences in trajectory arise from small changes in the aerodynamics early in the flight, and the differences in aerodynamic-coefficient estimates for the two codes are generally less than 5%. The altitude as a function of time for the cases is shown in Fig. 4 and shows again how the Aerodsn Mach 7.1 design flies higher and slower than the Missile Datcom missile. Figure 5 is a plot of the thrust as a function of time for each missile. The Mach 8.5 missiles start with about the same thrust, but the Aerodsn missile burns out with a greater final thrust. The lower-Mach-number cases have quite different thrust curves. The Missile Datcom thrust is much higher initially, but its curve is very flat, and the Aerodsn missile eventually achieves the maximum thrust value. The Mach number as a function of time is plotted in Fig. 6 for all cases and the curves show good agreement. The brief dips present in the curves are a result of temperature fluctuations within the atmosphere that affect the speed of sound and thus the Mach number. Figure 7 is a plot of the range as a function of time and is very similar for comparable cases.

Results from the four ballistic missile optimization runs demonstrate preliminary design level agreement between the aerodynamic codes. The maximum achievable ranges for cases with comparable Mach number limits are very similar regardless of the aerodynamic code used. In addition, the trajectory, thrust, and Mach number as a function of time are all very similar for comparable runs. The provided missile renderings demonstrate that typical missile designs were achieved with only small differences between the designs, mostly occurring in the size and placement of the fin sets. This can be seen in Figs. 8–11, which show the external configuration of each of the resulting missile designs, along with important performance data.

Although the performance plots indicate close agreement between the two different aerodynamic prediction routines at comparable

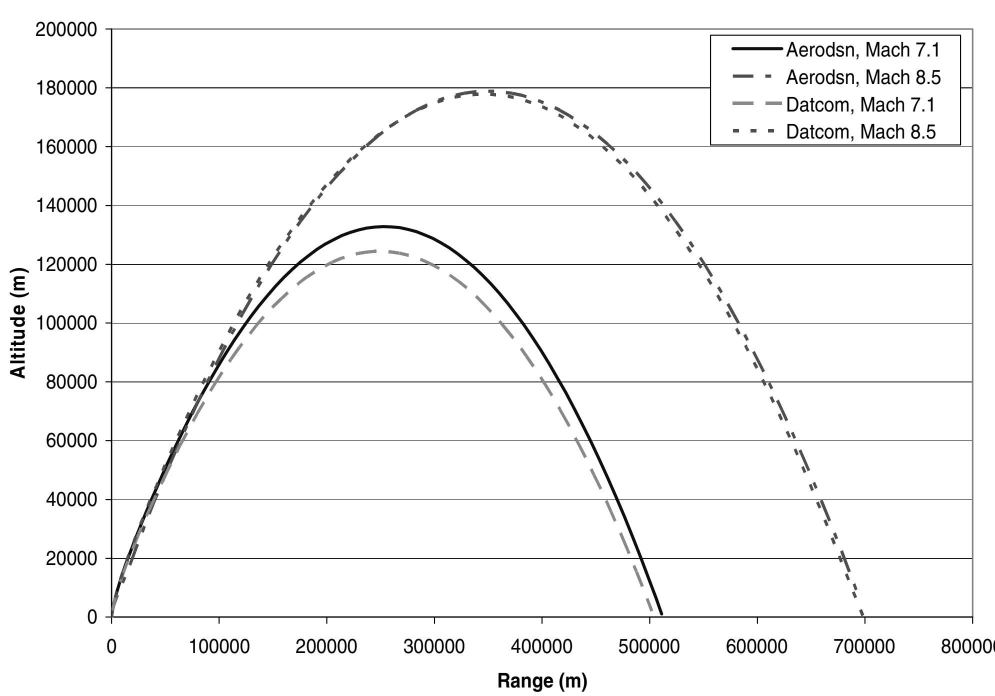

Fig. 3 Trajectory plot.

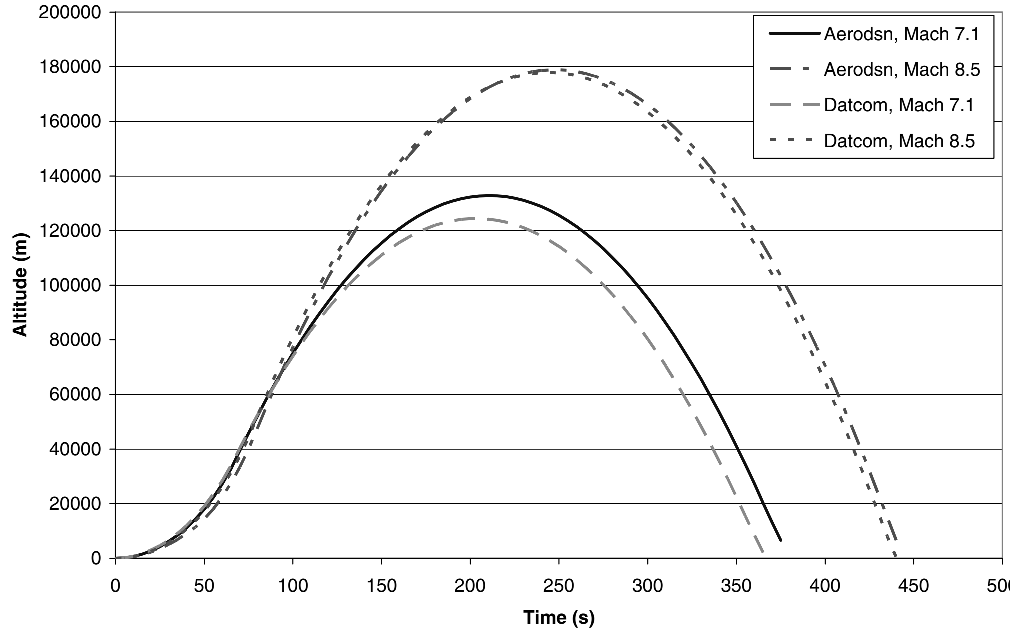

Fig. 4 Altitude as a function of time.

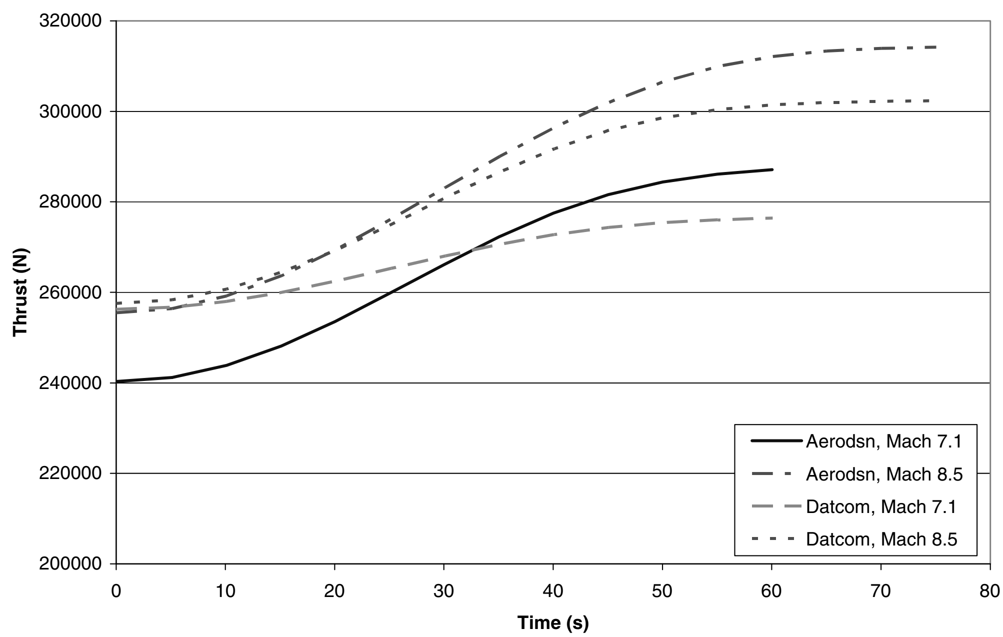

Fig. 5 Thrust as a function of time.

Mach number limits, these results alone cannot validate the aerodynamic prediction codes or even confirm that their predictions are similar, because much of the flight time for these ballistic cases occurs at very high altitudes. Because the missiles have no aerodynamic control, the primary aerodynamic force is drag, and above 10,000 m the drag force is very small. As shown in Fig. 4, these missile designs spend a considerable amount of time at altitudes at which the aerodynamic forces have little effect. For cruise missiles powered by airbreathing propulsion, the comparison and conclusion could be significantly different.

C. Guided Results

In addition to the ballistic optimizations, an aerodynamically controlled missile was optimized and the results are provided. The target was set at 400 km downrange and moving at 15 m/s in the y direction. The GA objective-function goals were set to minimize the miss distance, takeoff weight, and flight time and were given nearly equal numerical importance. The optimization was run with Aerodsn handling the aerodynamic analysis of the candidate missile designs, but the final optimized design was also flown with the Missile Datcom code to corroborate the results. Figure 12 shows the external configuration of the optimized missile design. Also listed in this figure are the results from the GA goals corresponding to the aeroprediction code that was used. Figure 13 is a plot from this optimization run of the convergence toward the GA goal.

The optimized missile is 16 m long with long slender fins placed as far aft as possible. At launch the missile has a mass just under 4800 kg. Aerodsn predicts that the missile will fly within 9 m of the moving target when it impacts the ground at 233 s. Candidate solutions were produced that hit much closer to the target, but at the cost of increased initial mass and a longer flight time. The GA goals could easily be reproportioned to place more importance on any single goal if it were deemed necessary. Missile Datcom shows good

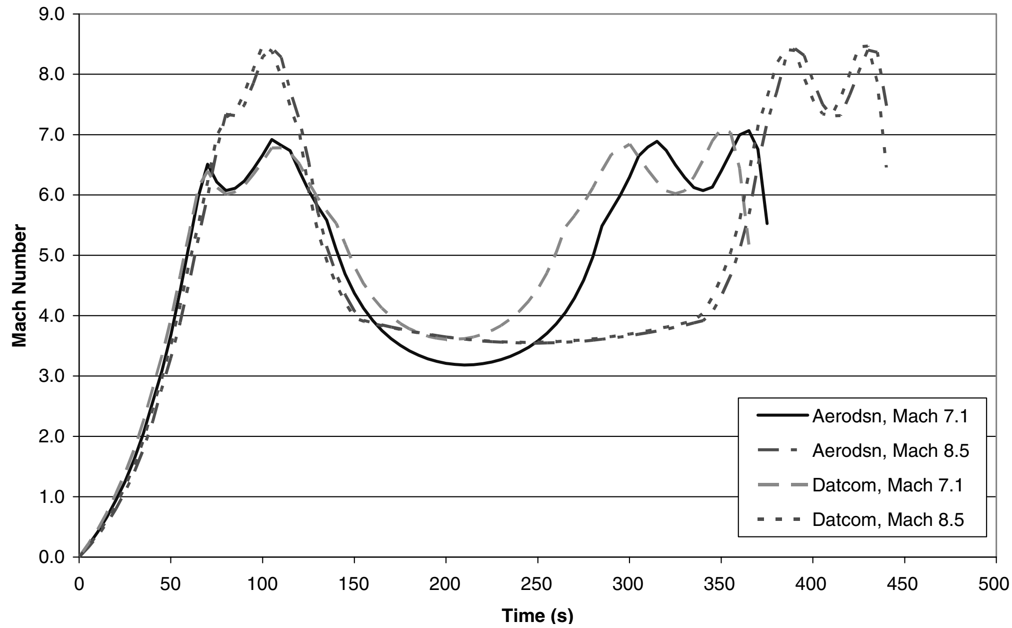

Fig. 6 Mach number as a function of time.

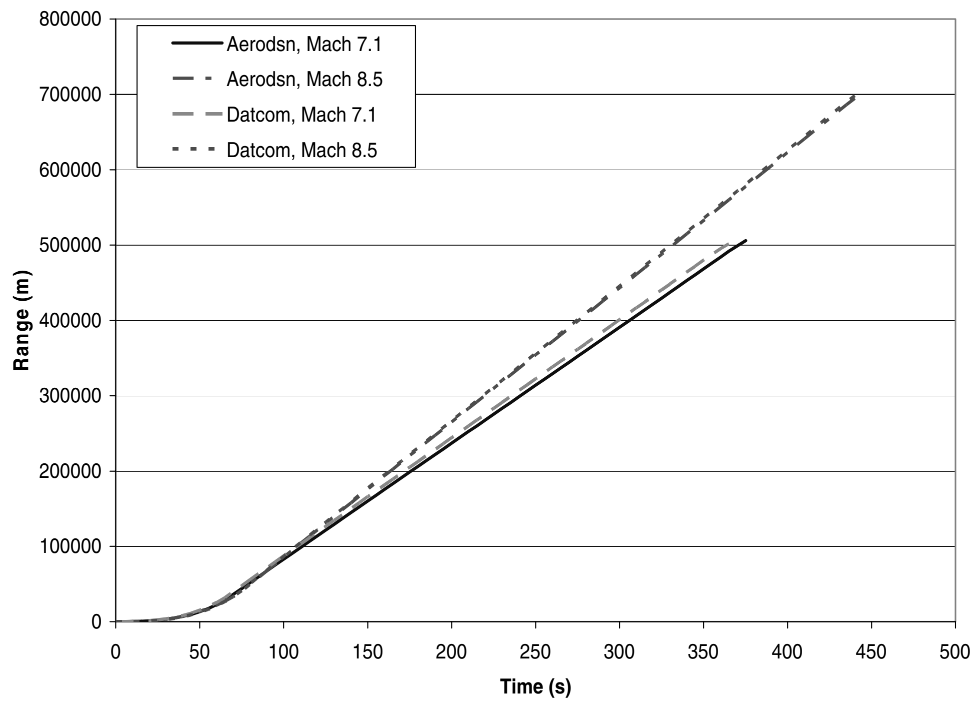

Fig. 7 Range as a function of time.

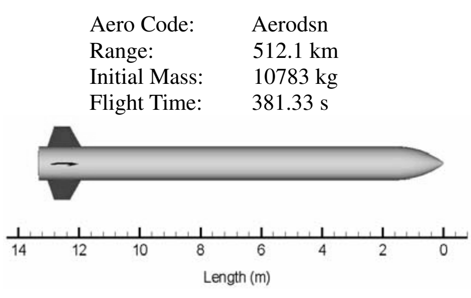

Fig. 8 Aerodsn Mach 7.1 missile.

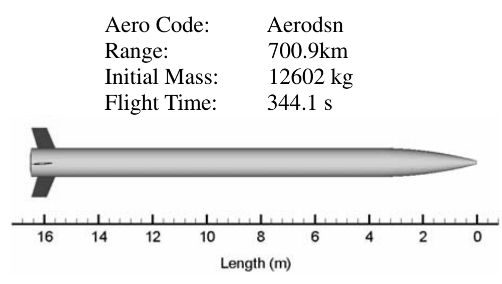

Fig. 9 Aerodsn Mach 8.5 missile.

Aero Code:	Missile Datcom
Range:	503.8 km
Initial Mass:	10712 kg
Flight Time:	366.9 s

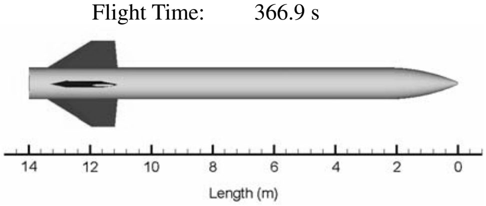

Fig. 10 Missile Datcom Mach 7.1 missile.

Aero Code:	Missile Datcom
Range:	698.1 km
Initial Mass:	12084 kg
Flight Time:	440.2 s

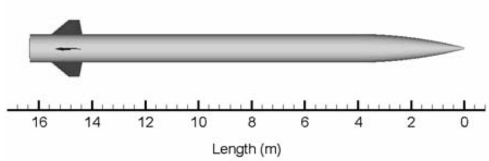

Fig. 11 Missile Datcom Mach 8.5 missile.

Aero Code:	Aerodsn	Missile Datcom
Miss Distance:	8.87 m	14.3 m
Initial Mass:	4785.3 kg	4785.3 kg
Flight Time:	233.4 s	231.9 s

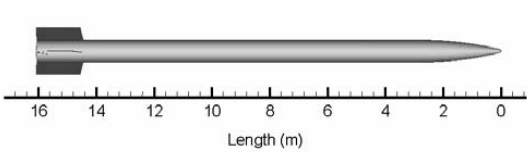

Fig. 12 Optimized missile external configuration.

agreement with Aerodsn's results, predicting that the same missile design would land 14.3 m from the target after flying 231.9 s. Trajectory plots for flights of the missile design using both aeroprediction codes are shown in Fig. 14. Missile Datcom predicts that the missile would take a lower altitude approach in comparison with the flight path that Aerodsn predicts the missile would take. This approach is consistent with the shorter flight time predicted by Missile Datcom; otherwise, the results are very similar.

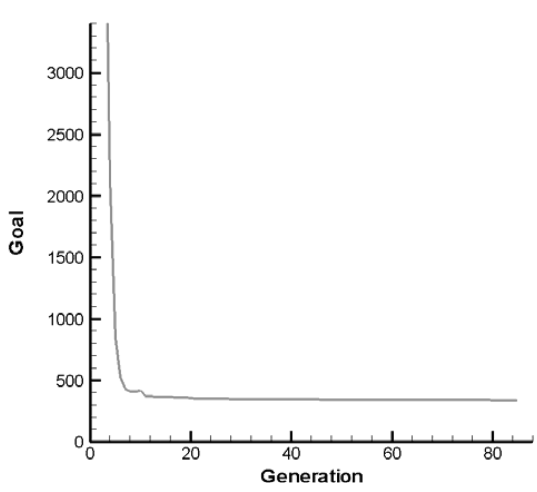

Fig. 13 Goal convergence.

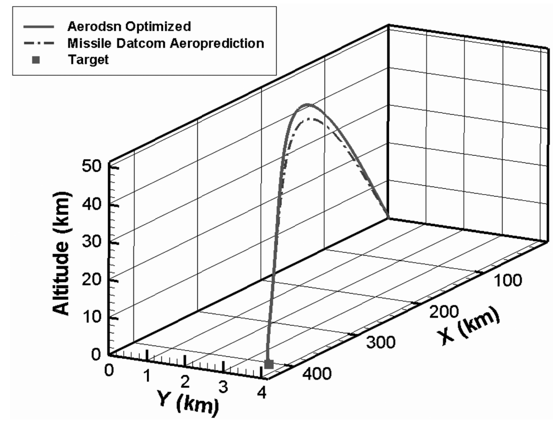

Fig. 14 Optimized missile trajectory.

IV. Conclusions

A comprehensive optimization tool for liquid-propellant missile systems was developed. The results represent a first of its kind in the published literature. This effort included a focus on one of the critical areas for missile design: aerodynamic prediction. Two different fast-aerodynamic-prediction codes, Aerodsn and Missile Datcom, were considered and compared for this function. The capability of the GA-based optimization code to successfully generate optimized preliminary designs of completely liquid-propelled missile systems was clearly demonstrated. The tool was shown to work with multiple goals and configurations. The two aerodynamic prediction routines produce slightly different results, but overall the codes are in good agreement.

References

[1] Burkhalter, J. E., Jenkins, R. M., and Hartfield, R. J., "Genetic Algorithms for Missile Analysis," Missile and Space Intelligence Center, Redstone Arsenal, AL, Feb. 2003.

[2] Hartfield, R. J., Burkhalter, J. E., Jenkins, R. M., and Metts, J., "Genetic Algorithm Upgrade Final Report," Missile and Space Intelligence Center, Redstone Arsenal, AL, 10 June 2005.

[3] Wollam, J., Kramer, S., and Campbell, S., "Reverse Engineering of Foreign Missiles via Genetic Algorithm," 38th Aerospace Sciences Meeting and Exhibit, Reno, NV, AIAA Paper 2000-0685, Jan. 2000.

[4] Holland, J., *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA, 1975.

[5] Metts, J., "Determination of Key Parameters for Reverse Engineering Solid Rocket Power Missiles," M.S. Thesis, Auburn Univ., Auburn, AL, Aug. 2006.

[6] Perhinschi, M. G., "A Modified Genetic Algorithm for the Design of Autonomous Helicopter Control System," AIAA Guidance, Navigation, and Control Conference, New Orleans, LA, AIAA Paper 97-3630, Aug. 1997.

[7] Karr, C. L., Freeman, L. M., and Meredith, D. L., "Genetic Algorithm Based Fuzzy Control of Spacecraft Autonomous Rendezvous," *Fifth Conference on Artificial Intelligence for Space Applications*, CP-3073, NASA, 1990.

[8] Mondoloni, S., "A Genetic Algorithm for Determining Optimal Flight Trajectories," AIAA Guidance, Navigation, and Control Conference and Exhibit, AIAA Paper 98-4476, Aug. 1998.

[9] Torella, G., and Blasi, L., "The Optimization of Gas Turbine Engine Design by Genetic Algorithms," 36th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, AIAA Paper 2000-3710, July 2000.

[10] Jang, M., and Lee, J., "Genetic Algorithm Based Design of Transonic Airfoils Using Euler Equations," 41st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA Paper 2000-1584, Apr. 2000.

[11] Jenkins, R. M., Hartfield, R. J., and Burkhalter, J. E., "Optimizing a Solid Rocket Motor Boosted Ramjet Powered Missile Using a Genetic Algorithm," 41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Tucson, AZ, AIAA Paper 2005-3507, July 2005.

[12] Anderson, M. B., Burkhalter, J. E., and Jenkins, R. M., "Design of an Air to Air Interceptor Using Genetic Algorithms," 1999 AIAA

Guidance, Navigation, and Control Conference, Portland, OR, AIAA Paper 99-4081, Aug. 1999.
[13] Anderson, M. B., "Design of a Missile Interceptor Using Genetic Algorithms," Ph.D. Dissertation, Auburn Univ., Auburn, AL, Dec. 1998.
[14] Perez, R. E., Chung, J., Behdinan, K., "Aircraft Conceptual Design Using Genetic Algorithms," 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, AIAA Paper 2000-4938, Sept. 2000.
[15] Schoonover, P. L., Crossley, W. A., and Heister, S. D., "Application of Genetic Algorithms to the Optimization of Hybrid Rockets," 34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, AIAA Paper 98-3349, July 1998.
[16] Burkhalter, J., Jenkins, R., Hartfield, R., Foster, W., Witt, J., and Heiser, M., "Missile Design Systems Developed with Genetic Algorithms," U. S. Army Aviation and Missile Command, Redstone Arsenal, AL, 31 Oct. 2001.
[17] Hartfield, R. J., Burkhalter, J. E., Jenkins, R. M., Metts, J. G., Riddle, D. B., and Dyer, J. D., "Genetic Algorithm Developments for Liquid Missile Analysis," Missile and Space Intelligence Center Rept. PAN 50353-05, Redstone Arsenal, AL, 22 Sept. 2006.
[18] Bailey, S. L., "Reverse Engineering of a SCUD Missile Using a Genetic Algorithm," M.S. Thesis, Auburn Univ., Auburn, AL, May 2004.
[19] Billheimer, J. S., "Optimization and Design Simulation in Solid Rocket Design," 3rd AIAA Solid Propulsion Conference, AIAA Paper 68-488, June, 1968.
[20] Anderson, M. B., Burkhalter, J. E., and Jenkins, R. M., "Multi-Disciplinary Intelligent Systems Approach to Solid Rocket Motor Design, Part 1: Single and Dual Goal Optimization," 37th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Salt Lake City, UT, AIAA Paper 2001-3599, July 2001.
[21] Blake, W. B., "Missile Datcom: User's Manual," Vol. 1, U.S. Air Force Research Lab., Wright–Patterson AFB, OH, 1998.
[22] Sanders, G. A., and Washington, W. D., "Computer Program for Estimating Stability Derivatives of Missile Configurations—Users Manual," U.S. Army Missile Command, Redstone Arsenal, AL, 1982.
[23] Etkin, B., and Reid, L. D., *Dynamic of Flight: Stability and Control*, Wiley, New York, 1996.
[24] Reynolds, W. C., *The Element Potential Method for Chemical Equilibrium Analysis Implementation in the Interactive Program STANJAN*, Dept. of Mechanical Engineering, Stanford Univ., Stanford, CA, Jan. 1986.
[25] Mattingly, J. D., *Elements of Gas Turbine Propulsion*, McGraw–Hill, New York, 1996.
[26] Sutton, G. P., and Biblarz, O., *Rocket Propulsion Elements*, Wiley, New York, 2001.
[27] "SS-1 'Scud' (R-11/8K11, R-11FM (SS-N-1B) and R-17/8K14)," http://www.janes.com/security/international_security/news/misc/sws_scud010426.shtml [retrieved 15 Nov. 2006].

M. Miller
Associate Editor