

DETERMINATION OF KEY PARAMETERS FOR REVERSE ENGINEERING  
SOLID ROCKET POWERED MISSILES

Except where reference is made to the work of others, the work described in this thesis is my own or was done in collaboration with my advisory committee. This thesis does not include proprietary or classified information.

---

Jonathan Glen Metts

Certificate of Approval:

---

John E. Burkhalter  
Professor Emeritus  
Aerospace Engineering

---

Roy J. Hartfield, Chair  
Associate Professor  
Aerospace Engineering

---

Winfred A. Foster  
Professor  
Aerospace Engineering

---

Stephen L. McFarland  
Dean  
Graduate School

DETERMINATION OF KEY PARAMETERS FOR REVERSE ENGINEERING  
SOLID ROCKET POWERED MISSILES

Jonathan Glen Metts

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama  
December 15, 2006

DETERMINATION OF KEY PARAMETERS FOR REVERSE ENGINEERING  
SOLID ROCKET POWERED MISSILES

Jonathan Glen Metts

Permission is granted to Auburn University to make copies of this thesis at its discretion, upon the request of individuals or institutions and at their expense. The author reserves all publication rights.

---

Signature of Author

---

Date of Graduation

## VITA

Jonathan Glen Metts, son of Glen S. Metts and Susan (Harris) Metts, was born on January 10, 1982 in Birmingham, Alabama. He lived with his family in Jasper, Alabama from birth until graduating from Walker High School in 2000. Jonathan attended Auburn University in fall 2000 until graduating magna cum laude with a Bachelor of Science in Aerospace Engineering in 2004. This period includes a summer spent abroad in St. Petersburg, Russia studying Russian language and culture. In the fall of 2004, he began graduate studies in the Auburn University Aerospace Engineering Department.

THESIS ABSTRACT

DETERMINATION OF KEY PARAMETERS FOR REVERSE ENGINEERING  
SOLID ROCKET POWERED MISSILES

Jonathan Glen Metts

Master of Science, December 15, 2006  
(B.S., Auburn University, 2004)

72 Typed Pages

Directed by Roy J. Hartfield

This thesis examines the process of reverse engineering solid rocket powered missiles using legacy codes and a genetic algorithm (GA). Available data for reverse engineering problems might include performance characteristics and internal and external geometric parameters. Reverse engineering from limited inputs, if proven to be practical and reliable, can be a critical means of obtaining more detailed information about missile programs from available data. For the present study, a baseline design for a solid rocket-propelled ballistic missile was reverse-engineered using a genetic algorithm and an accompanying set of programs designed specifically for solid rocket-powered missile prediction, including a six-degree-of-freedom dynamics simulator and solid propellant internal ballistics code. The goals were to match range, altitude, mass, and burnout time

of the baseline missile. Parameters being studied included propellant type, propellant radius ratio, fineness ratio, center body diameter, and nose length ratio. The objective of the GA and subsequent analyses was to find designs that closely matched the baseline model's performance characteristics and internal and external geometry. The focus of this thesis was to determine which combinations of design variables and other data should be known and to what precision, for a given confidence level in the reverse engineering solution.

## ACKNOWLEDGEMENTS

I would sincerely like to thank Dr. Roy Hartfield for offering me the research position which eventually led to the inspiration for this thesis. He has been remarkably patient, kind, and helpful. I would also like to thank Dr. John Burkhalter and Dr. Rhonald Jenkins for their work on the enormous software used in this thesis, as well as Dr. Murray Anderson for his original work on key portions of this software package.

Style manual or journal used

The American Institute of Aeronautics and Astronautics Journal

Computer software used

IMPROVE 3.1 Genetic Algorithm, General Purpose 6-DOF Simulation, MSIC-Final

Solid Rocket Missile Optimizer, AGARAT Data Analysis Tool, Compaq Visual Fortran,

WinDiff, TecPlot, Microsoft Excel, Microsoft Word



## TABLE OF CONTENTS

LIST OF FIGURES	x
LIST OF TABLES	xi
Introduction	1
Literature Review	4
Statement of Research Objectives	5
Methods and Materials	5
Setup for Test Cases	10
Results	16
Concluding Comments	54
Recommendations	55
REFERENCES	56
APPENDIX A: AGARAT Manual	60

## LIST OF FIGURES

Figure 1: Best Fitness Each Generation	18
Figure 2: Best Fitness Each Generation (Rescaled)	18
Figure 3: Reseed Fitness Progression	20
Figure 4: Altitude vs. Flight Time	21
Figure 5: Trajectories	23
Figure 6: Thrust Curves	25
Figure 7: Thrust Curves (Rescaled)	25
Figure 8: Mass Curves	27
Figure 9: Mass Curves (Rescaled)	28
Figure 10: All Propellant Grain Cross-Sections	30
Figure 11: External View of Best Matches	40
Figure 12: Altitude vs. Time of Flight with Dropped Mass Cases	48
Figure 13: Trajectories with Dropped Mass Cases	48
Figure 14: Thrust Curves with Dropped Mass Cases	49
Figure 15: Mass Curves with Dropped Mass Cases	49
Figure 16: Propellant Grain Cross-Sections with Dropped Mass Cases	51
Figure 17: External View of Dropped Mass Cases	53

## LIST OF TABLES

Table 1: Baseline Single Run Inputs	12
Table 2: Unlocked GA Parameters	13
Table 3: Studied Parameters and Locked Values	14
Table 4: Looper Binary Codes	15
Table 5: TOF Best Matches	21
Table 6: Best Range Matches	22
Table 7: Best Altitude Matches	22
Table 8: Initial Thrust Best Cases	24
Table 9: Zero Thrust Time Best Cases	24
Table 10: Primary Burn Ending Time Best Cases	24
Table 11: Initial Mass Best Cases	26
Table 12: Final Mass Best Cases	27
Table 13: Grain Cross-Section Best Cases	29
Table 14: Best Overall Matching Cases	39

## Introduction

Reverse engineering is a difficult and complex process even under optimal conditions, but the challenge is even greater when dealing with large numbers of design variables which may or may not be coupled to various performance parameters. Genetic algorithms offer some opportunities to learn about key aspects of a given design, using reverse-engineering results from very limited input data. Genetic algorithms may be capable of accomplishing this task in less time and with greater accuracy than many other methods, although nearly exact matches are not likely to be possible when only limited design parameter data is available.<sup>1-3</sup>

The GA uses the biological concept of generational adaptation to optimize a design problem which may contain numerous local optima. The IMPROVE 3.1 code employed in this investigation is based on the tournament method.<sup>4</sup> Rather than define the starting point for the optimization, the user inputs a range (maximum, minimum, and resolution) for each design parameter, and the GA randomly generates a population of candidate solutions from parameters within those ranges. After each candidate is analyzed by performance codes, the GA ranks the candidates (members) in order of fitness. Fitness is a measure of how well the member matches the objective function. Fitness is the GA's measure of quality in determining which members are best and should propagate throughout subsequent generations. The tournament method involves

randomly choosing two members from the population and picking the one which has the better fitness. Then this member is added to an intermediate population, from which the next real generation is created with mating (also known as crossover, in which two members exchange genetic material) and mutation processes. Over the course of many generations, the GA will find solution types that approach the target solution.

Limited past experience has shown that the success of this process depends upon which design parameters are known and how accurately they are constrained.<sup>1</sup> For a solid rocket missile, it is expected that the GA will not be able to find the correct internal propellant grain geometry without some *known* internal design parameter(s). Likewise, the GA may not be able to find the correct external missile geometry without one or more particular *known* external geometry parameters or details about performance. In the case of ballistic missiles, which are the subject of this thesis, the external geometry, including forward and aft control fins, is primarily responsible for flight stability. Because sufficient stability may be achieved by many different fin configurations, this aspect of the missile design is extremely difficult to match in a reverse-engineering process.

Multiple classes of answers from the GA may match the baseline performance. In a practical scenario, the baseline geometry may not be well known, so a reliable method of choosing the correct class of answers is required for the reverse engineering application to be useful. A tool has been developed to present full performance characteristics of multiple answers from the GA, so that results can, by inspection, be separated into families of similar answers. The correct family can then be identified based on which one most closely matches the known design configuration of the baseline model. (See APPENDIX A.)

A major advantage of the GA is that no initial solutions or guesses are required as input. This initial input would be difficult to provide for this type of problem, since a given performance goal could potentially be met by any number of missile designs which may differ significantly in size and shape. The GA is also flexible in how many and which design parameters can be constrained, although it is expected that constraining more design parameters will lead to more accurate results and that constraining particular parameters (for example, the fineness ratio of body length to missile diameter) is required to produce good results, whereas other parameters will be easily found by the GA and still others may not influence the performance goals enough to be accurately determined by the GA. This approach to reverse-engineering design of missiles has shown great promise; however, its success is limited in cases where certain key design variables are not well known.<sup>1</sup>

The aim of this research is to determine which set or sets of design variables must be known in order for the reverse-engineering process to successfully design the remainder of the missile. A comparison of the approach's success for varying GA goals will also be presented. The approach is illustrated by choosing a baseline design, for which all relevant design parameters and performance characteristics are known, and choosing various input configurations to determine which cases produce results closest to the baseline model.

A statistical approach to this problem may be worth pursuing in future research, in which statistical screening experiments would take a known set of parameter inputs (single-run case) and perturb selected parameters positively and negatively to measure the changes in the model's response (performance output).<sup>5</sup> Such an approach is

considerably impeded by the extreme nonlinearity of the problem and by the number and variety of equations used in the simulation to generate a response. Screening experiments have been performed on stand-alone solid rocket motor pairs, which have fewer design parameters than a complete missile.<sup>6</sup>

## Literature Review

The use of genetic algorithms in the field of aerospace engineering has rapidly expanded in the past decade. GA's have been applied to problems involving helicopters<sup>7</sup>, spacecraft controls<sup>8</sup>, flight trajectories<sup>9</sup>, gas turbines<sup>10</sup>, airfoils<sup>11</sup>, boosted ramjets<sup>12</sup>, interceptor missiles<sup>13</sup>, aircraft<sup>14</sup>, hybrid rockets<sup>15</sup>, liquid rockets<sup>1,16</sup>, solid rockets<sup>17,18</sup>, and other applications. Most of these applications involve forward design and optimization. Reverse-engineering in the aerospace field has also been attempted previously. Wollam, Kramer, and Campbell determined that a GA can find better reverse-engineering matches than a trial-and-error method, in the case of reverse-engineering air-to-air missiles with *a priori* propulsion data.<sup>3</sup> The design parameters for their report were non-dimensionalized to reduce dependency among parameters and result in fewer infeasible missile designs, an approach which has also been applied in this thesis. Their experiment was designed to prove the concept of a GA as a missile reverse-engineering tool, not to develop a detailed methodology of how to perform such an analysis.

Burkhalter, Jenkins, and Hartfield, in a report that directly inspired this thesis and used many of the same tools, found that a GA is capable of reverse-engineering the external geometry and some performance characteristics of a solid rocket powered

missile with a ballistic trajectory, but the internal propellant design is unlikely to be found without some initial knowledge of that design or detailed performance data.<sup>1</sup> Their report includes results for only four cases which are not directly comparable to one another because they have varying goals as well as varying input parameters. This thesis seeks to follow the same philosophy to obtain more complete, practical knowledge of the reverse-engineering methodology by comparing an exhaustive set of GA analyses (within certain assumptions and limitations required to reduce computation time) which differ only by selecting which design parameters are known initially.

### Statement of Research Objectives

This research sought to examine the relative importance of certain design parameters and combinations thereof when performing a reverse-engineering analysis on solid-rocket powered missiles with ballistic flight trajectories. This information can then be used to prescribe a methodology for the reverse-engineering process that will provide some consistency of success when applied to other cases in this class of missiles.

### Methods and Materials

The baseline model very closely approximates a real ballistic missile. All known parameters of the baseline have been entered into a forward-run simulation, and unknown parameters are extrapolated or estimated to complete the input set. The forward-run simulation uses the same rocket propulsion and missile design codes as the full GA, but it



simulates a single, known missile. The forward-run simulation produces the detailed performance characteristics, such as range, apogee, initial mass, and burnout time, as well as schematics of the baseline missile and its propellant grain. These results serve as the targets for the GA to match.

A series of GA analyses are then executed, each one testing the success of a different set of constricted (known) design parameters. Parameters not assumed to be known are allowed to fluctuate within a reasonable range of values. Due to the nonlinearity of the model, the GA should be able to find the correct value within the range for some parameters better than others. This experimental process generates a wide array of results which can be analyzed to determine which combinations work best. At the completion of each GA analysis, the best performers are compared to each other, so that the missile most closely matching the baseline may be found even though other missiles may have performed slightly better in terms of the GA goal. The GA ranks the best performers of each generation according to how well they minimize a weighted sum of errors in the following characteristics: flight range, maximum altitude (apogee), takeoff mass including propellant, and burnout time. The errors are the differences between the currently analyzed missile's characteristics and the corresponding characteristics of the baseline missile. Each error is non-dimensionalized by the baseline missile's characteristic value so that the four goals are of the same order of magnitude and can be properly weighted.

The final generation of each GA analysis is saved for later analysis. Also, a file is maintained to track the maximum, average, and minimum values of the objective function (weighted sum of the goals) across every generation of the GA analysis. The

data in this file can easily show whether the GA converged (trendwise) within the specified number of generations, and if so, how quickly that convergence occurred.

There are no strict convergence criteria.

The GA ranks missiles within each generation according to how well they minimize the characteristic equation. In a reverse-engineering application, this sorting method is sometimes too crude to determine the best match. A missile may perform slightly closer to the target characteristics but not be the best geometric match to the baseline missile, while another member slightly lower in the GA's sorting may be a superior geometric match and still perform nearly as well as the GA's so-called "best performer". Thus, the true best performer in a reverse-engineering problem should be chosen with the assistance of inspection, if possible, although the choice will sometimes coincide with the member top-sorted by the GA.

The best performer from each GA analysis is chosen to represent the level of success that particular GA analysis had in reverse-engineering the baseline missile. The pool of best performers across all GA analyses can thus be compared to determine which solutions best match the baseline case and, in the case of a tie or a near-tie, which ones converged in the fewest number of generations.

The genetic algorithm used is IMPROVE 3.1, (Implicit Multi-objective Parameter Optimization Via Evolution). This algorithm, written by Murray B. Anderson and based upon the original genetic algorithm theory developed by John Holland and others, optimizes a characteristic equation by generating and testing numerous members within a user-defined design space.<sup>4</sup> Many types of GA's have been developed. This GA uses the tournament method to sort population members. The first generation of population

members is randomly selected from within the allowed parameter ranges. Each member is assigned a fitness value based upon the response of the characteristic equation, and the GA sorts the members according to the results of this “tournament”. Then a new generation is formed by combining attributes of the best performers of the previous generation; some mutation factor(s) may also be applied to force the GA to explore more of the design space.

For this thesis, fifty-one generations of three hundred members each were analyzed for all thirty-two GA cases. Each GA case is a complete execution of the algorithm, simulating 15,300 missiles. Thus, a total of 489,600 missiles were tested, although not all of them were flown in the six degree-of-freedom (6-DOF) simulation due to geometric or other conflicts. The number of generations was chosen to give the GA ample opportunity to converge upon a good solution while minimizing the computation time required to produce such convergence in a majority of the cases. The number of members per generation is based upon a rule of thumb that this quantity should be roughly 1.6 times the number of bits to define the design space.<sup>2</sup> The design space varies from one case to another in this thesis, so the maximum of 192 bits was used as the basis. The corresponding number of 300 population members was then applied to all cases for computational uniformity. The number of cases executed depends on the number of parameters being analyzed, which as a power of two ( $2^5 = 32$  in this thesis) determines the number of cases needed to exhaustively analyze the possible cases. Increasing the number of parameters to switch between known/unknown values exponentially increases the required computation time as well as the complexity of results analysis. The thirty-

two cases executed in this thesis required a total of approximately three weeks of processing time on Intel Pentium 4, 2.8 GHz computers.

Because the GA forms successive generations from characteristics of the best performers of the previous generations, the overall quality of each generation tends to increase over the course of the analysis. In a reverse-engineering application, a match within acceptable error tolerances will almost certainly be found with large enough population sizes in the limit of infinite generations. However, the computing time is ultimately a major factor in such a simulation, so the number of generations must be chosen to strike a balance between ideal convergence of the solution and reasonable execution time for the program. One way to determine convergence is to look at the fitness of only the top-sorted member in each generation. With the “elitism” feature activated, the best performer is always preserved to the next generation to compete in the tournament with the other, new members. So if no new members are found which perform better than the previous best performer, it will be the best performer of the new generation as well. This feature ensures that the GA will always improve or maintain its best solution from one generation to another. It may take multiple generations to find a new best performer, so the maximum fitness value tends to jump upwards periodically throughout the GA analysis until near-convergence is achieved. Convergence may then be defined as a series of several generations in which the maximum fitness does not change; however, unless this solution is extremely close to the matching target, it should be stressed that the GA may eventually find a better solution after an unknown number of additional generations. Thus, convergence is an ambiguous topic as applied to the GA.

In this thesis, every GA analysis has been allowed to progress to the same number of generations, in order to provide a fair basis of comparison.

The fitness of each member is determined by a complex missile simulation external to the GA.<sup>1,2</sup> Each missile is defined by a set of design parameters, which are used to build up the missile's geometry and are combined with various material properties and other information which is assumed for the class of problem being optimized. There are subsequent modules to determine the propellant grain design, nozzle design, aerodynamic properties, and mass properties of the member. Using input from those modules, the missile is analyzed with a 6-DOF model, based on the dynamic equations of motion in both atmospheric and sub-orbital flight. The final results of the flight simulation in 6-DOF are passed back to the GA and are processed through the goal settings to determine the fitness of that member. This entire process is repeated thousands or even tens of thousands of times in a typical GA analysis.

### Setup for Test Cases

To determine parameter combinations required for successful reverse-engineering, the GA was set to match four key performance characteristics which may be determined from remote sensing of the baseline missile. Fitness in cases executed for this thesis is defined by the characteristic equation:

$$\begin{aligned} \text{Fitness} = & 3.0 * (\text{Range Goal}) + 2.5 * (\text{Apogee Goal}) \\ & + 2.0 * (\text{Mass Goal}) + 1.0 * (\text{Burnout Goal}) \end{aligned}$$

Where:

$$\text{Range Goal} = |(\text{actual range} - \text{target range})| / \text{target range}$$

$$\text{Apogee Goal} = |(\text{actual apogee} - \text{target apogee})| / \text{target apogee}$$

$$\text{Mass Goal} = |(\text{actual initial mass} - \text{target initial mass})| / \text{target initial mass}$$

$$\text{Burnout Goal} = |(\text{actual burnout time} - \text{target burnout time})| / \text{target burnout time}$$

Because there is no goal to define the accuracy of the missile's grain cross-section as matched to the baseline missile's grain cross-section, this important result for reverse engineering is not factored into the GA sorting and must be determined by inspection.

The overall matching accuracy of the thrust curve cannot be determined from the burnout time alone, so the best thrust curve match must also be determined by inspection.

Because the objective function is a weighted combination of the four goals listed above, the best matches for each individual goal must also be determined by inspection.

The baseline missile data was drawn from technical specifications of a real missile, although some parameters had to be estimated. A single-run simulation of the baseline missile was completed in order to produce the baseline performance output data.

An external view of the baseline missile is shown in Figure 11.

Table 1: Baseline Single-Run Inputs

Propellant Type:	4.000	Wing Taper Ratio:	0.0001
Propellant Radius:	0.5409	Wing LE Sweep Angle (deg):	0.0001
Inner Radius:	0.2076	Wing LE X-Location / Body Length Ratio:	0.3500
Number of Points/Spokes:	5.0000	Tail Fin Exposed Semi-Span Length:	0.7495
Fillet Radius:	0.0566	Tail Fin Root Chord Length:	1.6865
Epsilon:	0.8194	Tail Fin Taper Ratio:	0.7777
Point / Spoke Angle (deg):	36.5159	Tail Fin LE Sweep Angle (deg):	0.0100
Fractional Nozzle Length:	0.7500	Tail Fin LE X-Location / Body Length Ratio:	0.9155
Throat Diameter	0.2749	Autopilot Delay Time (sec):	9000.0000
Nose Radius Ratio:	0.0100	Autopilot Time Constant (sec):	0.0001
Nose Length Ratio:	4.3097	Autopilot Damping:	0.0001
Total Body Length (Fineness Ratio):	13.3038	Crossover Frequency (Hz):	0.0001
Center Body Diameter (in.):	31.1024	Pronav Gain:	0.0001
Wing Semi-Span Length:	0.0001	Initial Launch Angle (deg):	75.0000
Wing Root Chord Length:	0.0001		

Note: All lengths except Center Body Diameter are non-dimensionalized by dividing by the Center Body Diameter.

The basic GA input file has the 29 design parameters set to the maximum and minimum reasonable values. This file was then modified for each of the 32 cases being analyzed. However, there are many possible combinations of design parameters which are reasonable taken independently but which will not produce a reasonable missile. For this reason, conflicts have been preprogrammed so that non-workable designs will be

rejected before analysis and assigned penalty values for the goals. This process occurs more frequently in early generations, before better performing genetic material permeates the population.

Table 2: Unlocked GA Parameters

<b>Parameter</b>	<b>Maximum</b>	<b>Minimum</b>	<b>Resolution</b>
Propellant Type	8.0000	1.0000	1.0000
Propellant Outer Radius Ratio	0.9500	0.1000	0.0100
Propellant Inner Radius Ratio	0.9500	0.1000	0.0100
Number of Points / Spokes	17.0000	3.0000	2.0000
Fillet Radius Ratio	0.3000	0.0010	0.0010
Epsilon	0.9900	0.1000	0.0100
Point / Spoke Angle (deg)	85.0000	15.0000	1.0000
Fractional Nozzle Length Ratio	0.9000	0.6000	0.0100
Throat Diameter	0.5000	0.1500	0.0010
Nose Radius Ratio	0.2500	0.0100	0.0100
Nose Length Ratio	5.0000	1.0000	0.0100
Total Body Length (Fineness Ratio)	25.0000	2.0000	0.1000
Center Body Diameter (in.)	35.0000	1.0000	0.1000
Wing Exposed Semi-Span Length	2.0000	0.0100	0.0100
Wing Root Chord Length	5.0000	0.0100	0.0100
Wing Taper Ratio	5.0000	0.0100	0.0100
Wing LE Sweep Angle (deg)	45.0000	1.0000	1.0000
Wing Axial Position to Body Length Ratio	0.5000	0.1000	0.1000
Tail Exposed Semi-Span Length	2.0000	0.1000	0.1000
Tail Root Chord Length	5.0000	0.1000	0.1000
Tail Taper Ratio	5.0000	0.1000	0.1000
Tail LE Sweep Angle (deg)	45.0000	1.0000	1.0000
Tail Axial Position to Body Length Ratio	0.9000	0.6000	0.0100
Auto Pilot Delay Time (sec)	9000.1000	9000.0000	0.1000
Auto Pilot Time Constant (sec)	0.0020	0.0010	0.0010
Auto Pilot Damping	0.0020	0.0010	0.0010
Crossover Frequency (Hz)	0.0020	0.0010	0.0010
ProNav Gain	0.0020	0.0010	0.0010
Initial Launch Angle (deg)	89.0000	30.0000	0.1000

Note: All auto-pilot parameters are locked such that the auto-pilot mode will not be engaged, because only ballistic missiles are considered.



Table 3 contains the locked parameter values that were switched on and off in different combinations to produce the 32 cases of the experiment. These parameters were chosen based on *a priori* knowledge of which design parameters most strongly affect the flight performance of a solid rocket propelled ballistic missile.<sup>1</sup> Each case is assigned a binary code indicating which of the five parameters, if any, are locked. (Examples: 00000 has no locked parameters; 10101 has locked values for propellant type, fineness ratio, and nose length ratio.) The binary system fits the investigation setup because each studied parameter has one of two states: locked or unlocked. The short binary strings show, at a glance, how much input data was known in a given case. Note that this binary sequence is completely unrelated to the binary system that determines a particular member's gene alleles in the GA proper.

Table 3: Studied Parameters and Locked Values

<b>Binary Pos.</b>	<b>Parameter</b>	<b>Maximum</b>	<b>Minimum</b>	<b>Resolution</b>
1	Propellant Type	4.1000	4.0000	0.1000
2	Prop. Outer Radius Ratio	0.5410	0.5400	0.0010
3	Fineness Ratio	13.3100	13.3000	0.0100
4	Body Diameter (in.)	31.1100	31.1000	0.0100
5	Nose Length Ratio	4.3100	4.3000	0.0100

The simple “Looper” program shell was designed to sequentially and automatically execute the 32 cases; however, upon learning that each case required approximately 12 hours to complete, the Looper program was slightly modified into multiple versions which would skip ahead to certain cases. These different versions could then be executed on multiple computers, effectively parallelizing the process.

The Looper program uses the five-digit binary sequence to determine which of the 32 cases to analyze. This program customizes the GA input file to lock down certain parameters to the values in Table 3 depending on the assigned binary sequence, as explained above. The entire GA suite is then executed normally, but afterwards, the final population file (population.00050, which is actually the fifty-first generation because the first is called generation 0), is saved under a unique filename so that data is not lost during the next case. The GA's fitness progression file (ga\_out1.dat) is also saved under a unique filename.

Table 4: Looper Binary Codes

Case #	Binary Code	Case #	Binary Code
1	00000	17	10000
2	00001	18	10001
3	00010	19	10010
4	00011	20	10011
5	00100	21	10100
6	00101	22	10101
7	00110	23	10110
8	00111	24	10111
9	01000	25	11000
10	01001	26	11001
11	01010	27	11010
12	01011	28	11011
13	01100	29	11100
14	01101	30	11101
15	01110	31	11110
16	01111	32	11111

After all the cases were executed, the final generation of each case was analyzed using the Advanced Genetic Algorithm Results Analysis Tool (AGARAT) software to determine the best reverse-engineering match by inspection. AGARAT picks out a

number of the top-sorted members in the population file and executes each one as a single-run simulation, saving the performance output and internal and external geometric data under unique filenames. For this thesis, the top five members (as sorted by the GA) were analyzed. Customized Microsoft Excel ® macros were used to generate plots to compare the five members and the baseline. The best reverse-engineering match can then be chosen based on these plots. The grain cross-section and thrust and mass curves are given priority in the inspection process, because the portions of these results were not programmed as GA goals.

After the true best match is chosen for all 32 cases, these members are compared to each other and the baseline case in plots generated by additional Excel ® macros. These plots indicate which cases resulted in the best reverse-engineering solution. By looking at the locked-unlocked parameter sequence for the most successful cases, a survey of the critical parameters and general reverse-engineering strategy can be formulated.

## Results

Although the GA fitness progression plots do not always give a completely accurate portrayal of the reverse-engineering results, because they rely only on the GA's sorting and not on inspection, they do provide a general idea of how well the experiment worked. Note that the objective function was minimized, so the best cases are those with fitness values closest to zero. Figure 1 shows all 32 cases, indicating the stratification caused by varying the known parameters. Some cases are clearly far more successful

than others, at least based on the minimum/best fitness of each generation. Figure 2 is a more detailed plot of only the ten best cases, or those with the lowest-valued objective functions. Analysis of this plot shows that, of the ten cases displayed, six have at least three locked parameters, and nine have at least two locked parameters. The tenth, Case 5, apparently found an excellent member by random chance and, due to the elitism feature, carried it through the remaining generations. Eight of the ten best cases have a locked fineness ratio (total missile length over center body diameter), indicating that this parameter may be critical to the reverse-engineering process. Six of the ten best cases have a locked propellant radius variable, which factors into the grain design and partially determines the thrust profile and propellant grain mass. The remaining three parameters being studied are locked in exactly half of the top ten best cases.

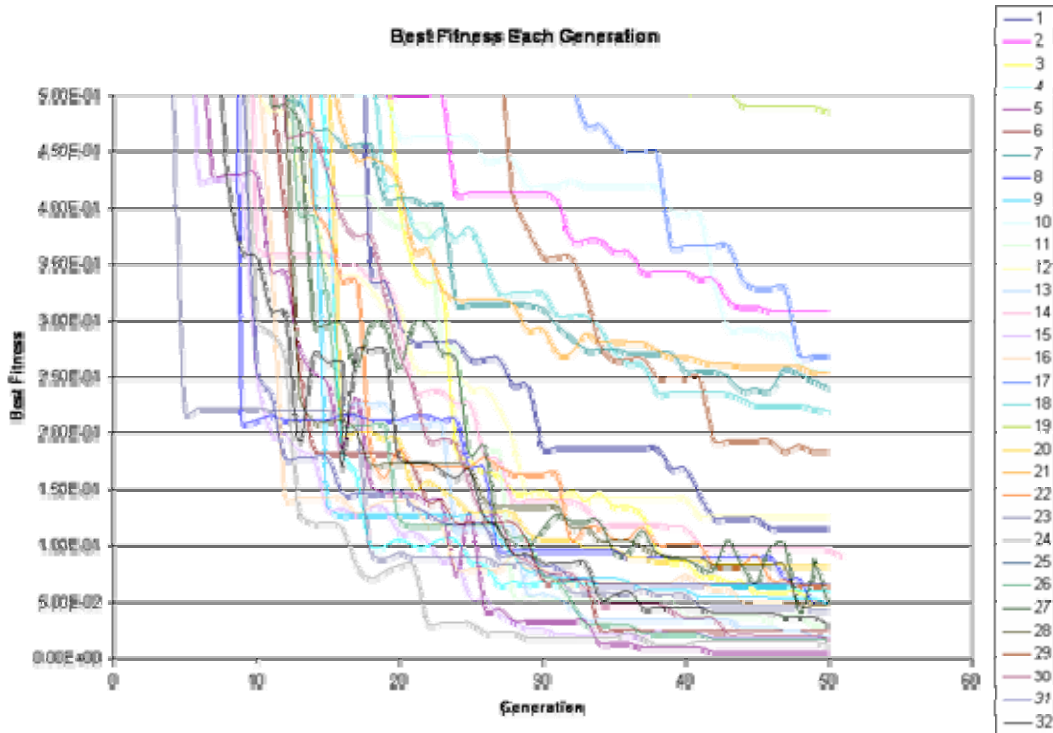


Figure 1: Best Fitness Each Generation

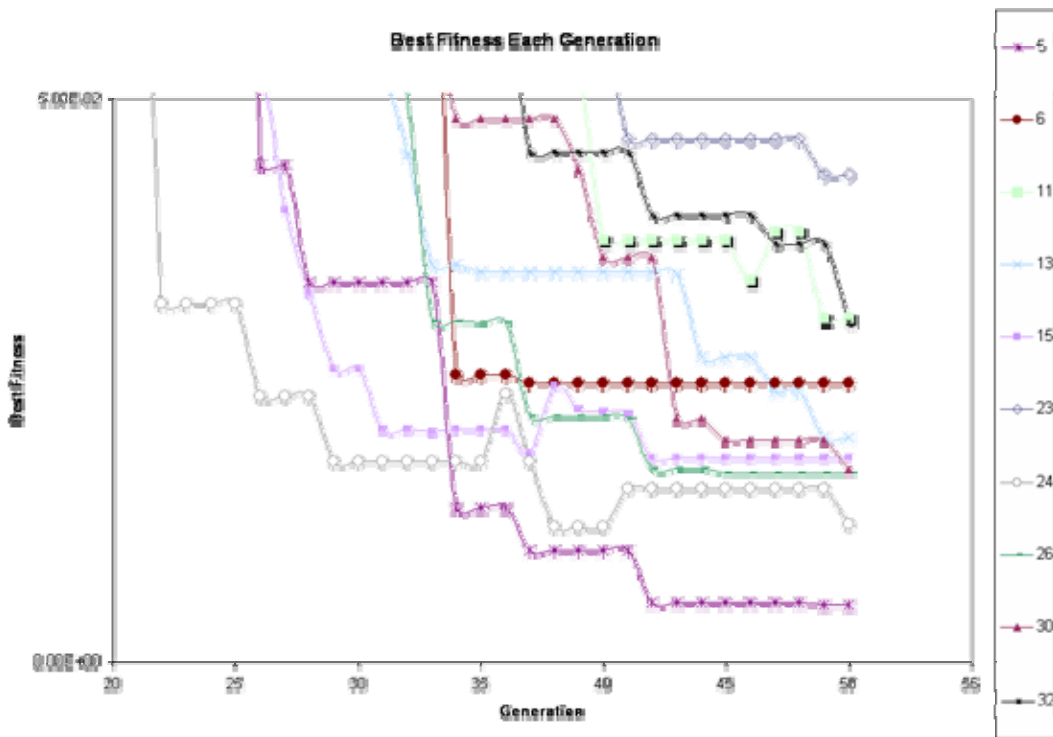


Figure 2: Best Fitness Each Generation (Rescaled)

To illustrate the effect of randomness in the GA, Cases 5 and 32 were analyzed again with a different random number seed. The fitness progression for these cases is compared with that of the original cases in Figure 3. (Note that the vertical axis uses a logarithmic scale.) The unusually good performance of the original Case 5 is contrasted with the reseeded Case 5, which performs more normally. The fact that Case 32 performs similarly before and after reseeding shows the result, noted by Wollam, Kramer, and Campbell, that reseeding typically does not cause significant changes in the fitness of solutions found by the GA after an equal number of generations.<sup>3</sup> It must be stressed that in a real application of the GA, random fitness improvements such as the one seen in Case 5 are in fact helpful coincidences; the abnormality is only a detriment to this particular investigation because it confuses the case-to-case comparisons designed to analyze sensitivity to design parameters.

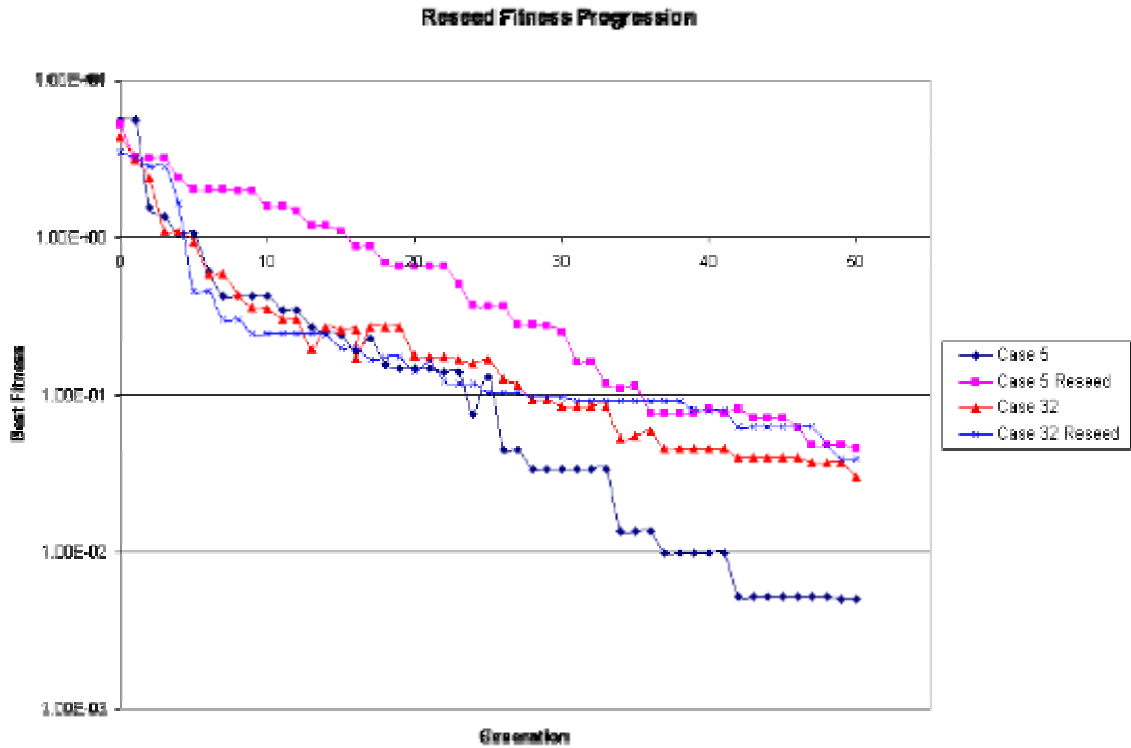


Figure 3: Reseed Fitness Progression

While analysis of the GA fitness progression plots may give preliminary indications of the results sought by this experiment, a more valid and thorough analysis must be performed after the inspection of each case for the true best reverse-engineering match. The following series of figures displays those results.

In the plot for time of flight (TOF), the series most closely matching the baseline performance are shown in Table 5. The TOF plot also shows altitude, but that characteristic is paired with range for the trajectory plot and will be examined later in this section.

Table 5: TOF Best Matches

Case #	Binary
7	00110
5	00100
28	11011
14	01101
25	11000
29	11100
9	01000
32	11111
13	01100
30	11101

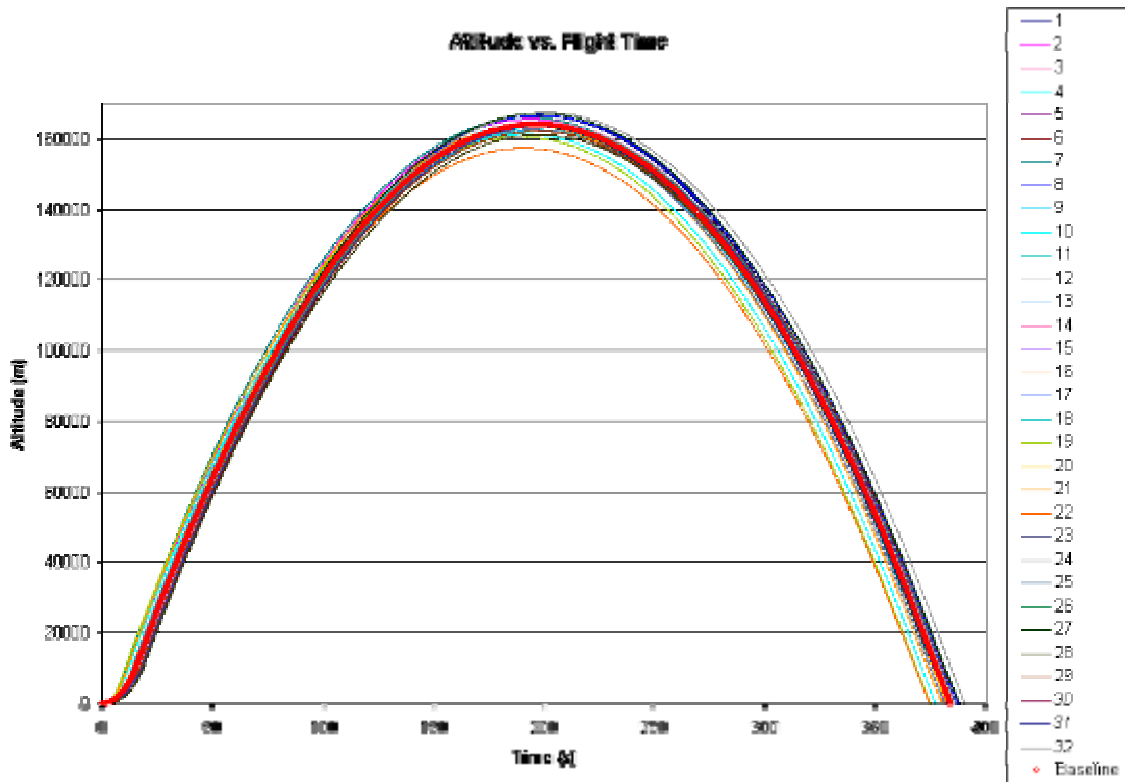


Figure 4: Altitude vs. Flight Time

In the trajectory plots shown in Figure 5, the series most closely matching the baseline for range and altitude are given in Tables 6 and 7. All of the cases match the



baseline trajectory reasonably well, which is expected because the range and altitude goals are most heavily weighted in the objective function.

Table 6: Best Range Matches

Case #	Binary
18	10001
13	01100
12	01011
25	11000
31	11110
16	01111
11	01010
26	11001
2	00001
3	00010

Table 7: Best Altitude Matches

Case #	Binary
12	01011
6	00101
30	11101
21	10100
20	10011
16	01111
17	10000
13	01100
31	11110
26	11001

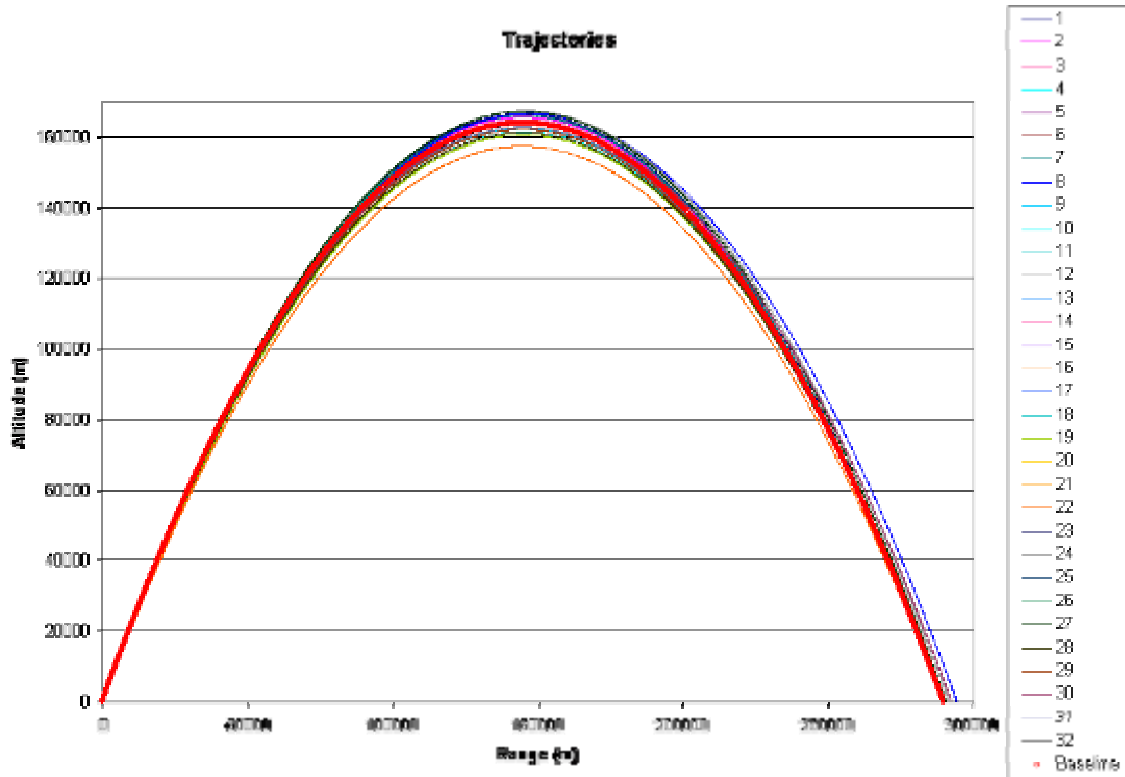


Figure 5: Trajectories

Figure 6 shows all the thrust curves, and Figure 7 contains the same data but is rescaled, and the poorest cases are hidden. Note that total impulse, which is the area under the thrust curve, is relatively constant across all the cases. Three characteristics of the thrust curves are used to determine the best match: initial thrust, time at zero thrust, and time at end of primary burning. The latter is the point at which the thrust curve drops sharply. The best cases for each of these characteristics are found in Tables 8 through 10. Also, note that in Table 8, bolded/italicized cases are those which exhibit progressive burning slopes similar to that of the baseline thrust curve.

Table 8: Initial Thrust Best Cases

Case #	Binary
<b>13</b>	<b>01100</b>
9	01000
<b>11</b>	<b>01010</b>
6	00101
<b>1</b>	<b>00000</b>
<b>32</b>	<b>11111</b>
<b>5</b>	<b>00100</b>
<b>27</b>	<b>11010</b>
<b>15</b>	<b>01110</b>
31	11110

Table 9: Zero Thrust Time Best Cases

Case #	Binary
32	11111
15	01110
26	11001
4	00011
29	11100
28	11011
31	11110
24	10111
12	01011
27	11010

Table 10: Primary Burn Ending Time Best Cases

Case #	Binary
14	01101
16	01111
32	11111
13	01100
30	11101
5	00100
27	11010
8	00111
1	00000
17	10000

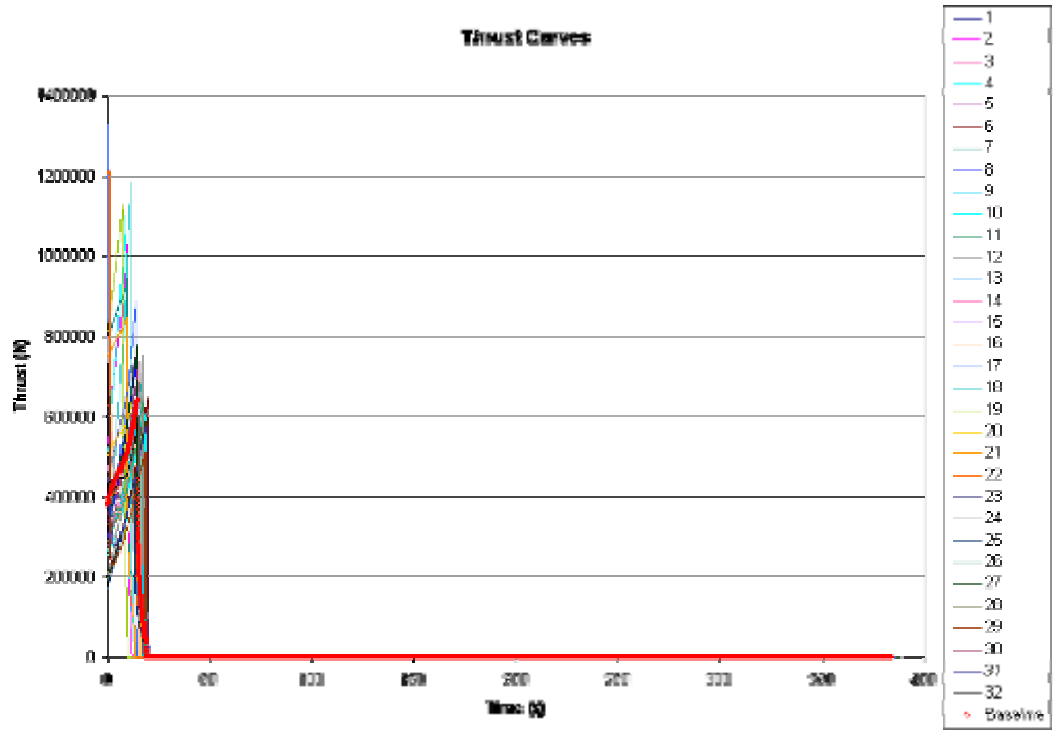


Figure 6: Thrust Curves

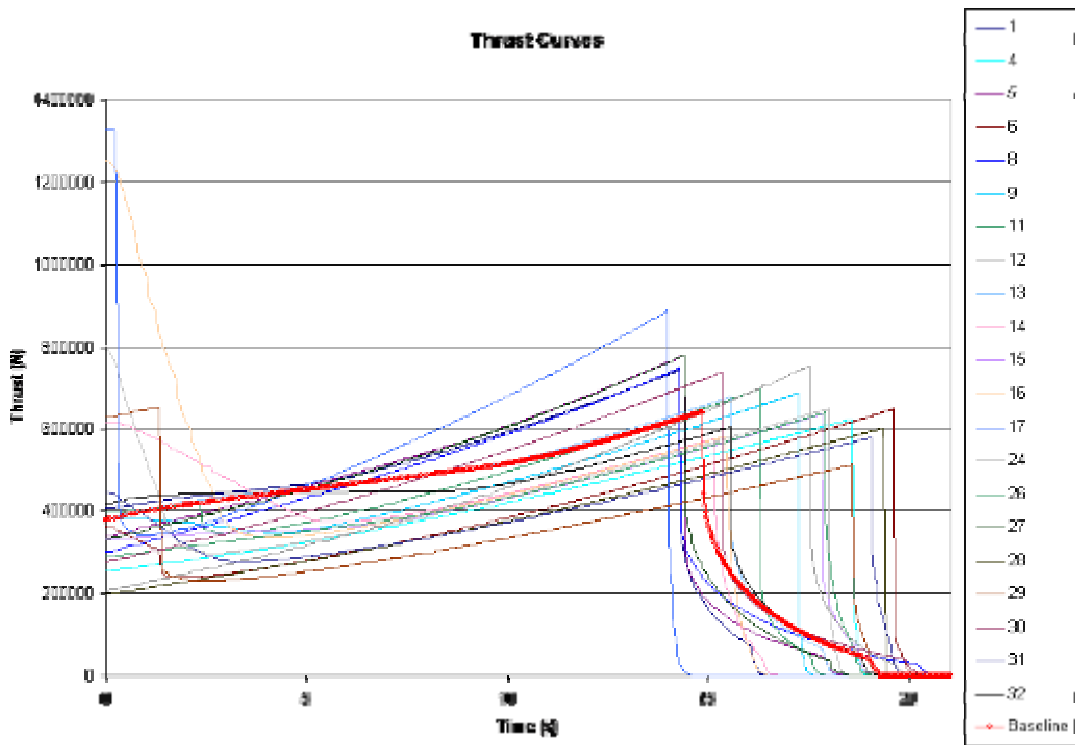


Figure 7: Thrust Curves (Rescaled)

In comparing the mass profiles among all the cases, two characteristics were used: initial mass and final mass. The best cases in each category are listed in Tables 11 and 12. The mass ratio, defined as the ratio of propellant mass to total mass, is also shown in these tables. For reference, the baseline missile has a mass ratio of 0.622. The mass-time curves are shown in Figures 8 and 9, the latter re-scaled to show increased detail during the first 25 seconds of flight, before the propellant burns out and the mass becomes constant. The poorest cases have been dropped from Figure 9 to make the results more readable.

Table 11: Initial Mass Best Cases

Case #	Binary	Mass Ratio
20	10011	0.636
3	00010	0.666
26	11001	0.636
5	00100	0.643
11	01010	0.628
1	00000	0.655
30	11101	0.634
24	10111	0.624
15	01110	0.649
13	01100	0.667

Table 12: Final Mass Best Cases

Case #	Binary	Mass Ratio
24	10111	0.624
12	01011	0.643
8	00111	0.616
16	01111	0.618
2	00001	0.623
18	10001	0.615
32	11111	0.623
11	01010	0.628
6	00101	0.634
30	11101	0.634

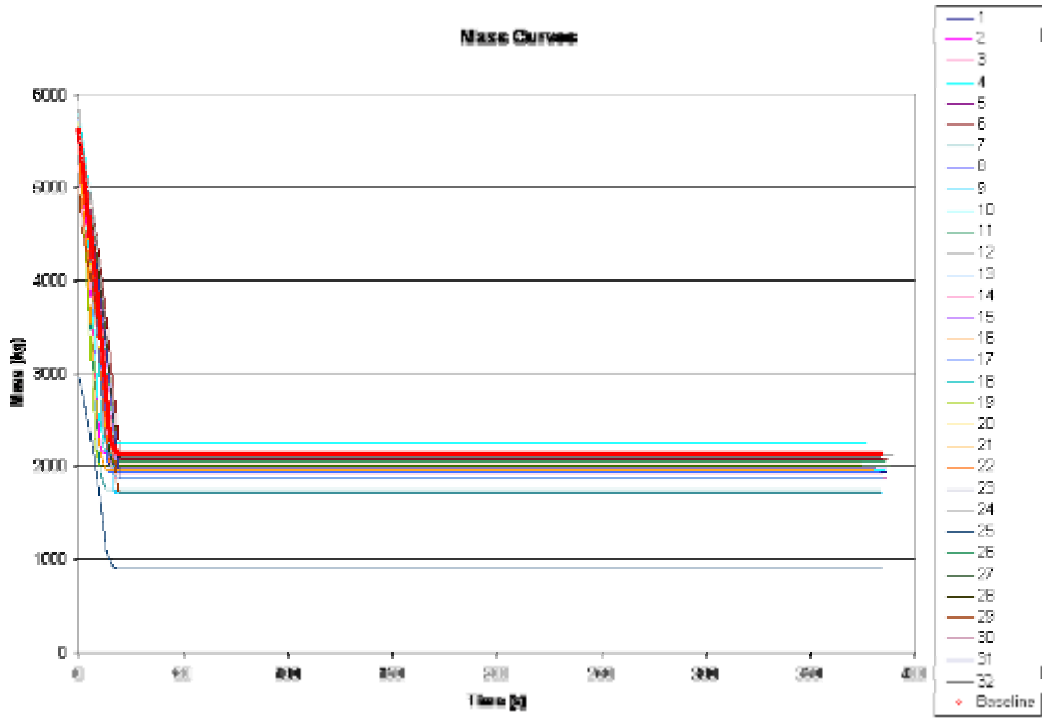


Figure 8: Mass Curves

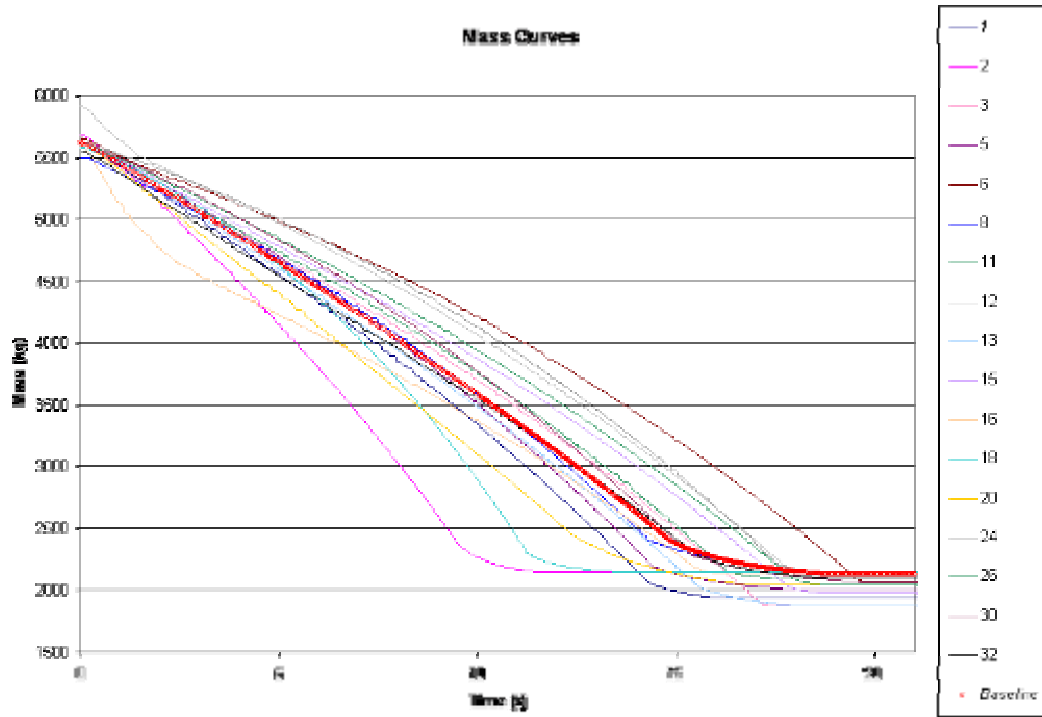


Figure 9: Mass Curves (Rescaled)

In Figure 10, the cross-sectional propellant grain profiles of all cases were plotted and compared to the baseline grain. The inner-most line corresponds to the initial propellant cross-section. The remaining contour lines indicate the shape of the grain at various intervals during the burn, until burning eventually reaches the missile casing, which is the outside circle. There is no discrete set of characteristics for the grain matching comparison, but a good match should have roughly the same port (hollow) area and initial perimeter as the baseline, and the correct number of star points is also desirable. Excellent matches for the grain cross-section were not expected in this investigation, due to the complex nature of grain geometry and the limited amount of information given to the GA. Only one internal geometry parameter was among the set of five studied parameters, and none of the fitness goals directly relate to the grain

geometry. Nevertheless, several good matches were found by the GA and are listed in Table 13. In particular, Cases 13 and 32 are very close matches.

Table 13: Grain Cross-Section Best Cases

Case #	Binary
32	11111
13	01100
1	00000
20	10011
27	11010
7	00110
5	00100
15	01110



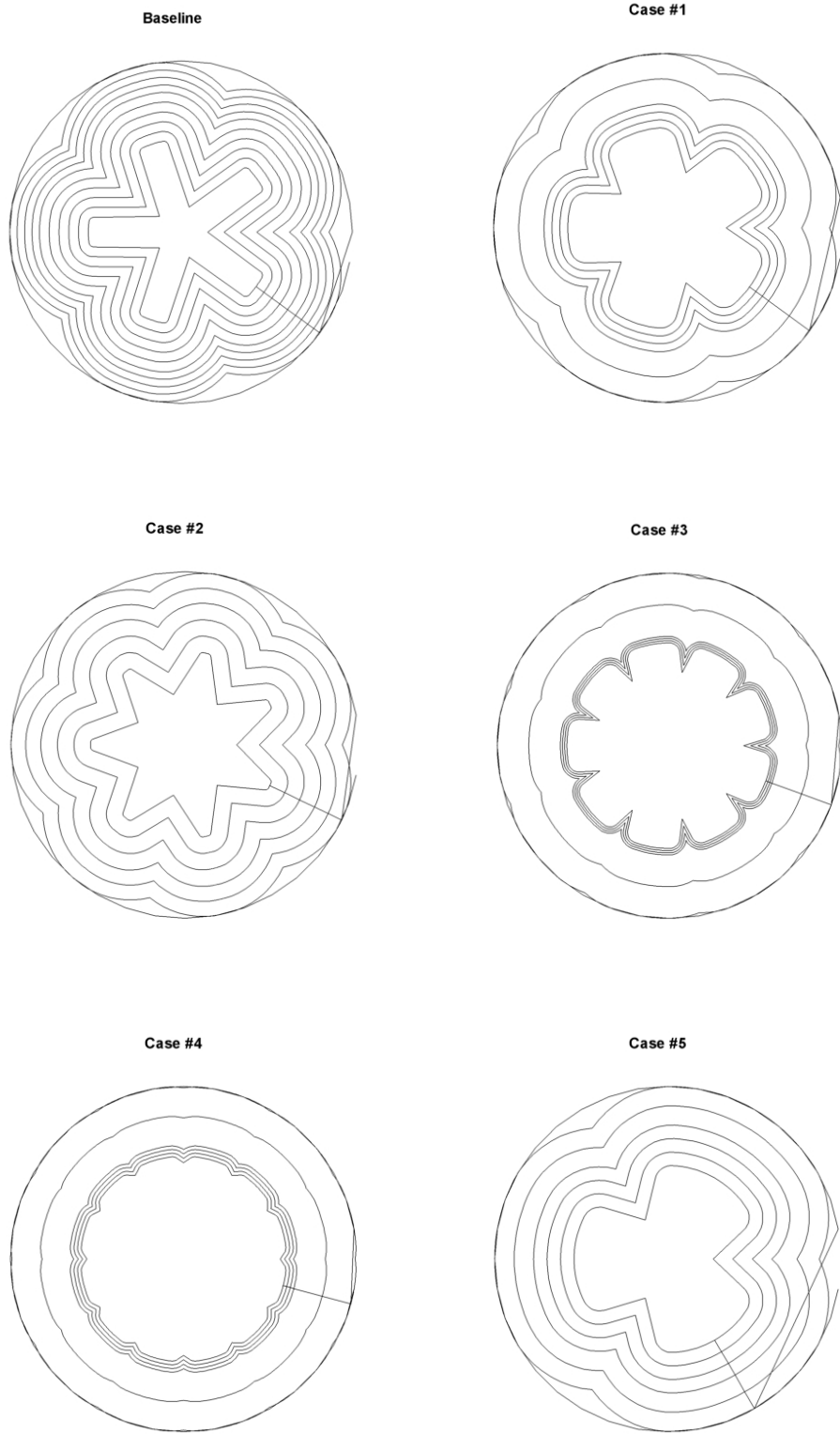
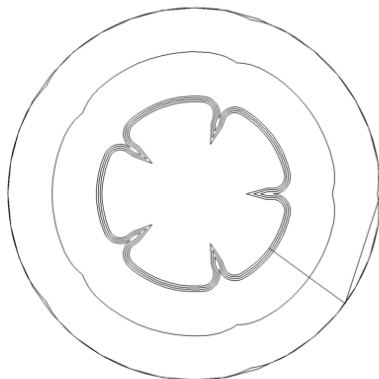
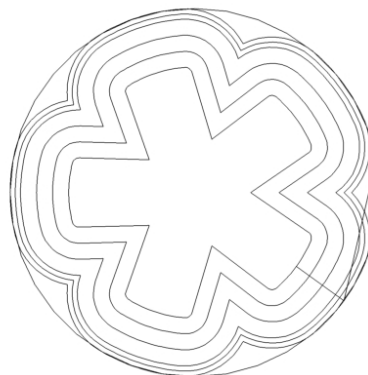


Figure 10: All Propellant Grain Cross-Sections

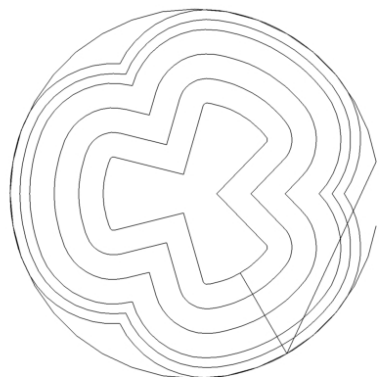
Case #6



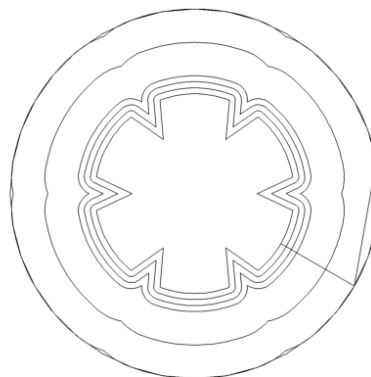
Case #7



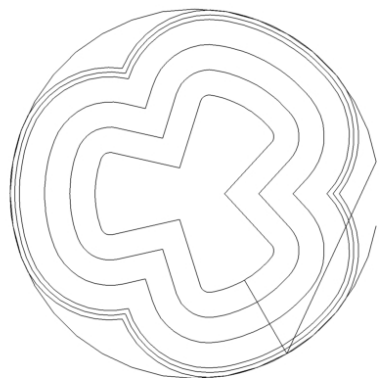
Case #8



Case #9



Case #10



Case #11

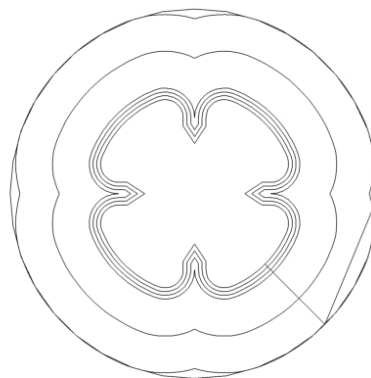
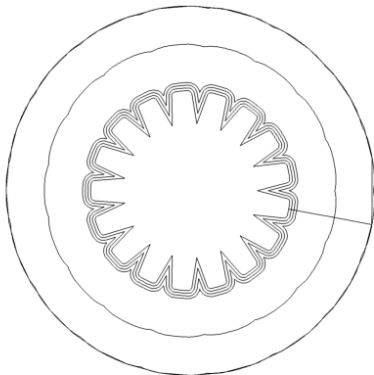
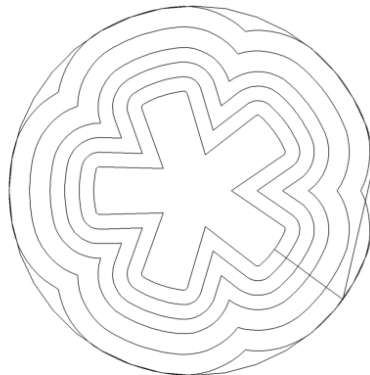


Figure 10: All Propellant Grain Cross-Sections (cont.)

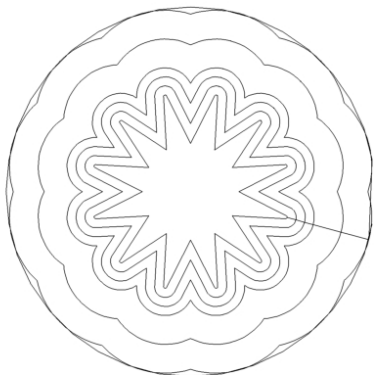
Case #12



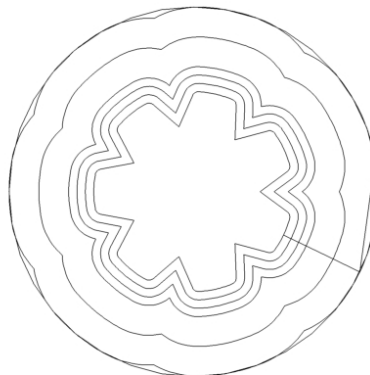
Case #13



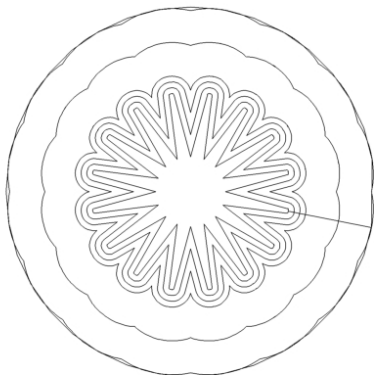
Case #14



Case #15



Case #16



Case #17

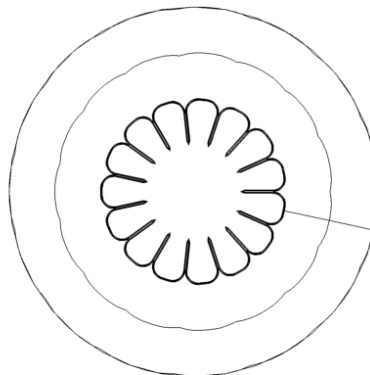
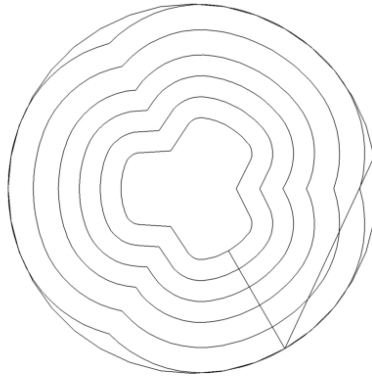
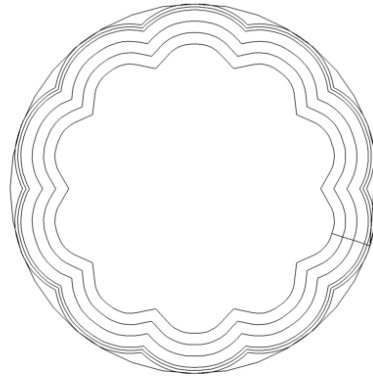


Figure 10: All Propellant Grain Cross-Sections (cont.)

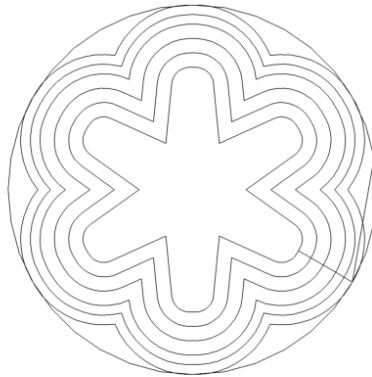
Case #18



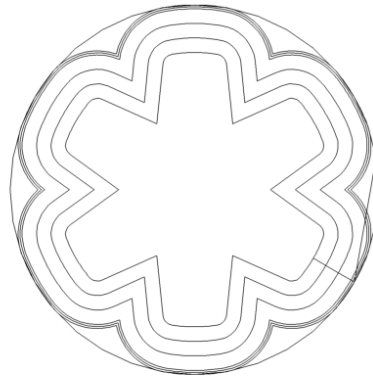
Case #19



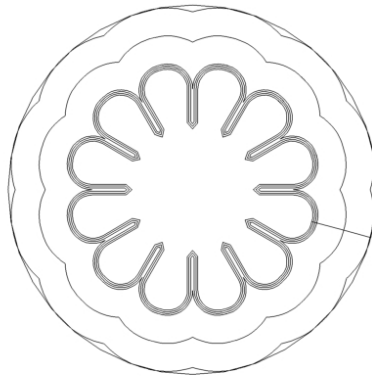
Case #20



Case #21



Case #22



Case #23

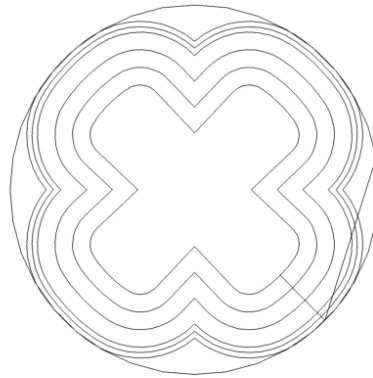
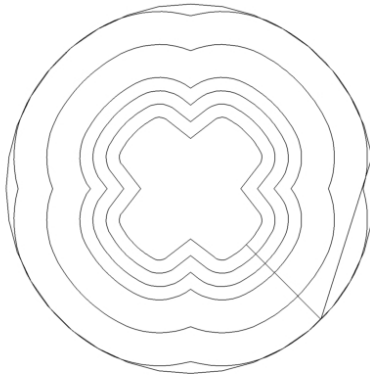


Figure 10: All Propellant Grain Cross-Sections (cont.)

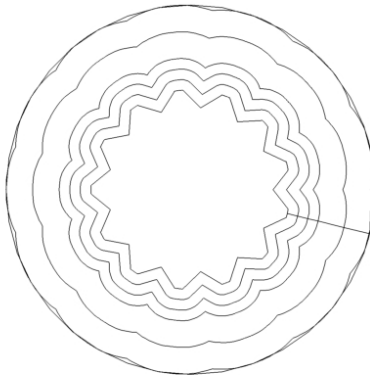
Case #24



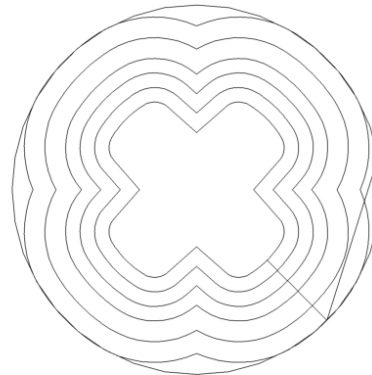
Case #25



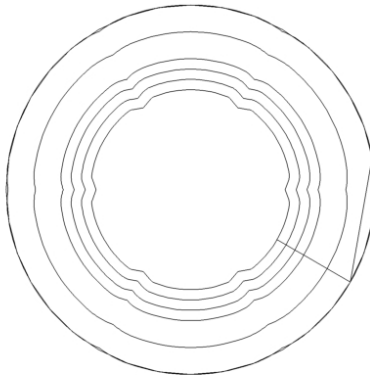
Case #26



Case #27



Case #28



Case #29

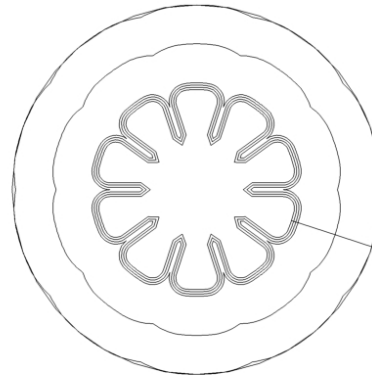


Figure 10: All Propellant Grain Cross-Sections (cont.)

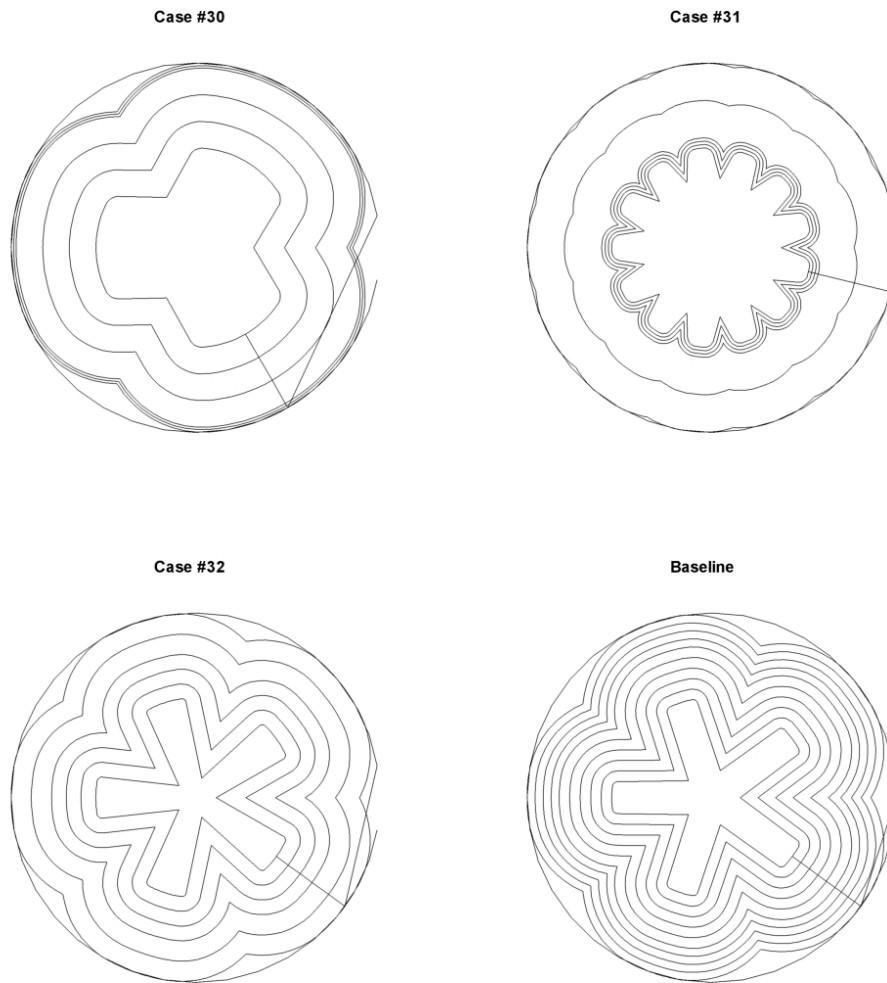


Figure 10: All Propellant Grain Cross-Sections (cont.)

There is a significant relationship between the best grain cross-section matches and the best thrust curve matches. Among the characteristics examined in the thrust curves, Cases 5, 13, 15, and 31 closely match two of the baseline curve's three characteristics. Cases 27 and 32 match all three of the characteristics. Note that all of these cases which match the baseline thrust curve so well also exhibit grain geometries similar to the baseline grain geometry and are therefore listed in Table 13. This

correlation is significant because the thrust curve can be extrapolated from radar and other surveillance data, while the internal grain geometry cannot. Thus, using the GA to match known thrust curve characteristics can offer valuable information about the corresponding unknown propellant grain geometry, assuming that the grain is of the same family. The advantage of executing multiple GA cases is also depicted in this example; although none of the best thrust curve matching cases are exact matches for the grain geometry, the similarities among these cases correspond very well with the baseline grain. A comparison of these cases shows that most of them have large star points protruding deep into the center of the grain. This description also closely matches that of the baseline grain geometry. The number of star points varies, but a plurality (three of the six cases) have five points, which is the same number as in the baseline grain. A caveat for these cross-referenced cases which correlate thrust curve accuracy to grain accuracy is that all of them except Case 5 have the locked value for propellant radius, which is an internal parameter. While this study has not shown that external data can help the GA to match the internal missile characteristics, it does at least indicate that knowing some information about the internal configuration can help the GA to find much more internal data.

Among the various tables of best matches shown above, a few cases appear in more than half of the tables, indicating that these cases are the best overall matches. Analysis of the parameter binary strings for these cases shows some trends. Among the twelve cases that appear in at least four of the nine tables, all but two have a locked value for the non-dimensional propellant radius ratio. All the best cases except Case 1 (which had no locked parameters at all) have locked values for propellant radius ratio, fineness

ratio, or both. The other three parameters under study are locked and unlocked evenly among the best cases. Although Case 1 is not the strongest match, it performed very well considering that none of the parameters for this case were locked. The success of this case shows the power of the GA to direct solutions towards the correct configuration if the objective function is properly designed and the goal matching characteristics are well known. It should be noted that the matching targets for the four fitness goals are considered to be known facts about the baseline to be reverse-engineered; thus, even with no locked parameters, Case 1 had four key pieces of performance information to aid in finding the correct solution. The widely varying success of other cases, all of which had more known information than Case 1, indicates that while knowing certain pieces of information can help the GA find a better match, it is also possible to over specify the problem in that detailed information about one or two parameters used at GA inputs can, in some cases, lead to reduced success in extracting other parameters. The overall best matching case, Case 13, which appears in seven out of nine tables of best matches in various categories, has locked values for the aforementioned parameters: propellant radius ratio and fineness ratio. Other cases which lock additional parameters find slightly inferior matches, including Case 32, which has all five parameters locked. Case 32 is certainly one of the best overall matches, however, as are most cases which have four locked parameters. This trend indicates that locking as many parameters as possible is generally a valid strategy. Consciously not locking certain parameters, even if the true value is known, can occasionally result in a better match, but the issue of which parameters to leave unlocked can be very complex. The seemingly illogical possibility of the GA finding a better solution with less input data is due to the random nature of the



algorithm. If these cases were extended to an arbitrarily large number of generations, which is outside the scope of this thesis, the random factors would likely diminish in effect and cases with more locked information might more consistently converge to the best matches.

Deciding which parameters to lock should also be influenced by the chosen fitness goals. Within the table of best matches in each category, some localized trends are evident. Most of the best matches for range and initial thrust have propellant radius ratio locked. Fineness ratio is locked in most of the best matches for phase one end time. Propellant radius ratio and fineness ratio are virtually tied for frequency among the best matches for time of flight and altitude. Body diameter and propellant type are both common among the best matches for burnout time, while nearly all of the best matches for final mass have the nose length ratio locked. The remaining categories, including grain geometry, showed no clear trends among the five studied parameters.

Table 14 compiles the cases which appear in at least four of the previous tables; these cases are good matches in multiple areas of comparison and thus may be considered the overall best matches found by the GA. Figure 11 shows the external geometry for the baseline missile and the overall best cases with 2D views generated in TecPlot ®. The aerodynamic coefficients required to match a ballistic flight trajectory may be achieved by many fin configurations. Therefore, good matches for the fin sizes and locations are not expected without providing some data about the fin geometry, which was not done for this investigation. The external views do allow for some examination of the length scales of solutions found by the GA. Two of the studied parameters, fineness ratio and nose length ratio, correspond to the missile's axial length. Consequently, the best

matches for external geometry tend to be those cases which have locked values for these two parameters.

Table 14: Best Overall Matching Cases

Case #	Binary	# of Criteria Matched Well
1	00000	4
5	00100	5
11	01010	4
12	01011	4
13	01100	7
15	01110	4
16	01111	4
26	11001	4
27	11010	4
30	11101	5
31	11110	4
32	11111	6

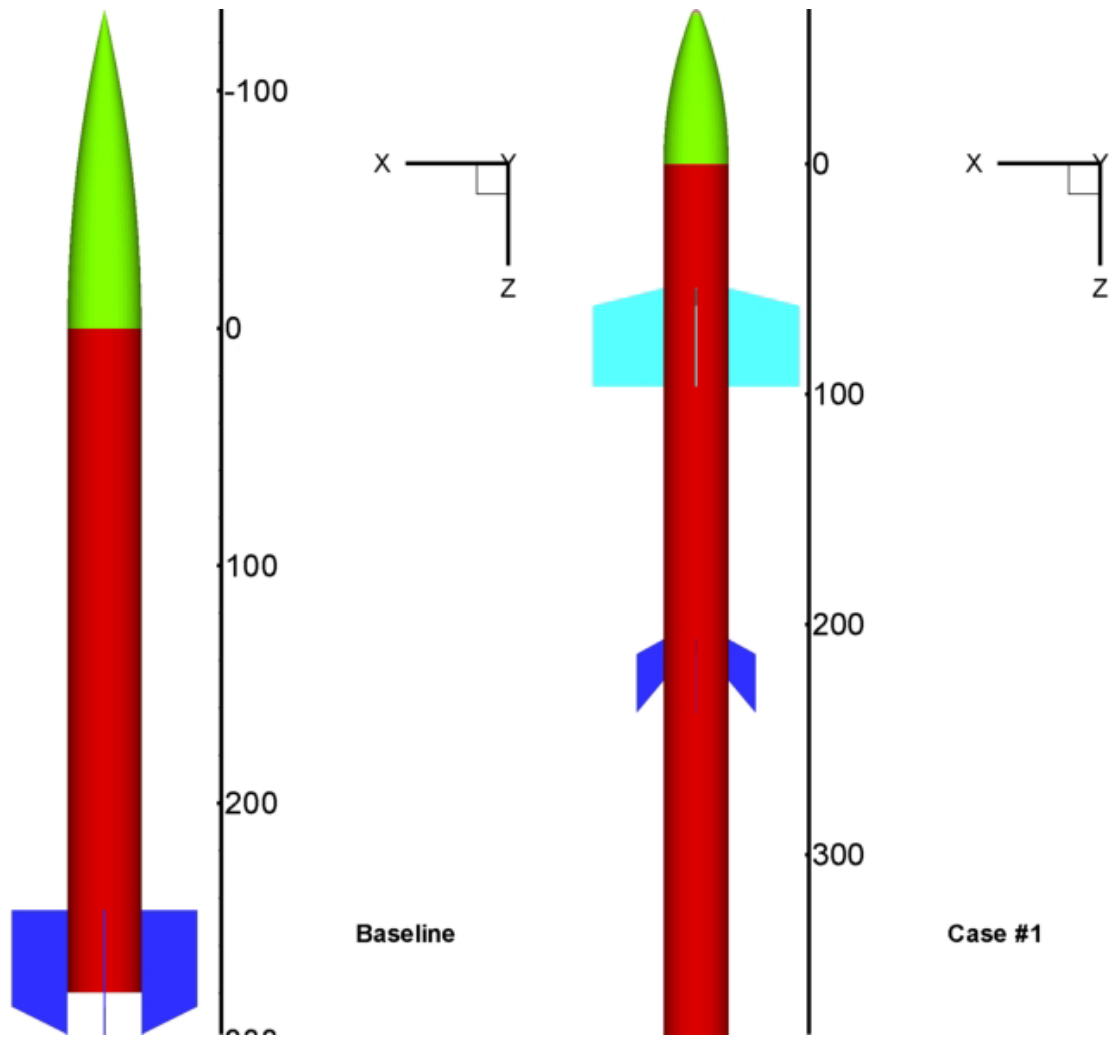


Figure 11: External View of Best Matches

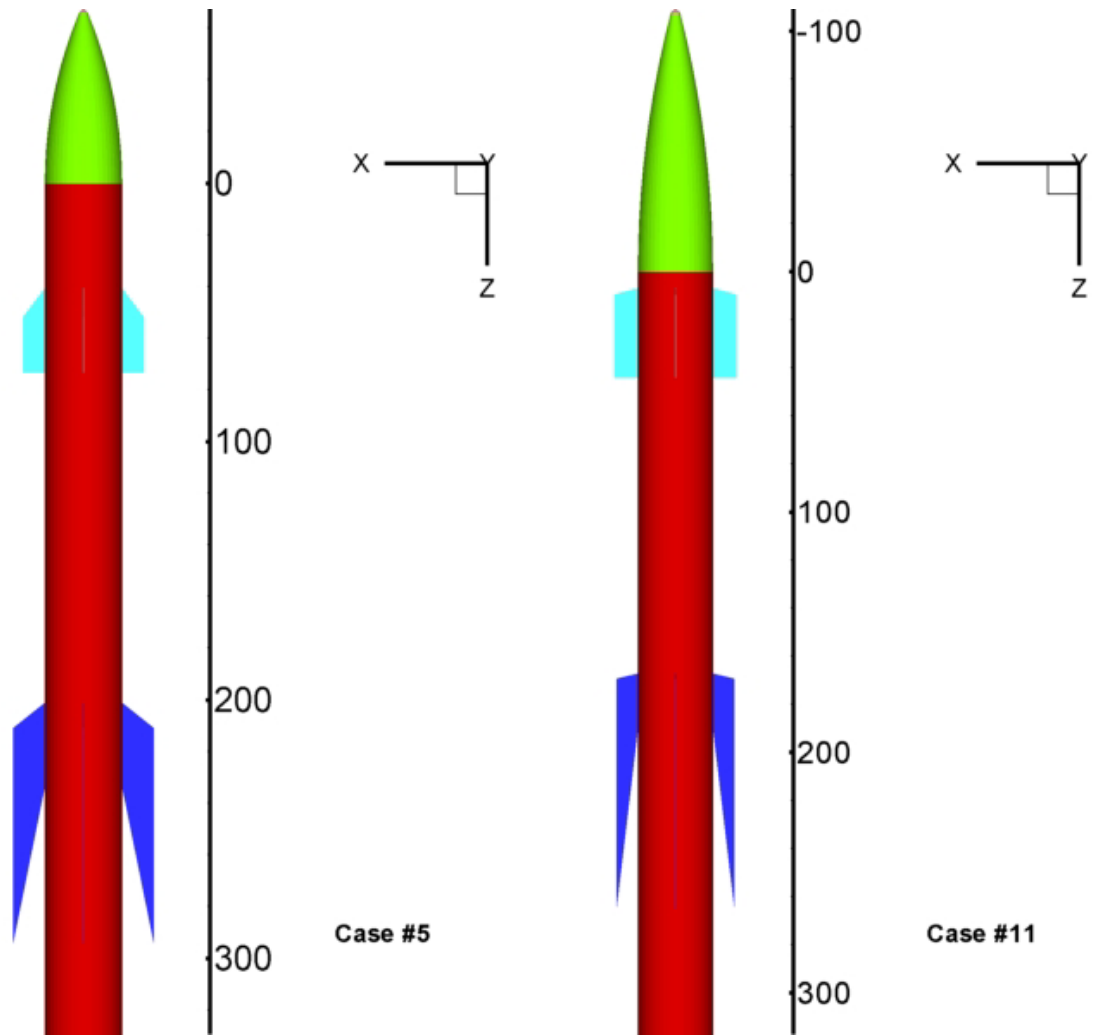


Figure 11: External View of Best Matches (cont.)

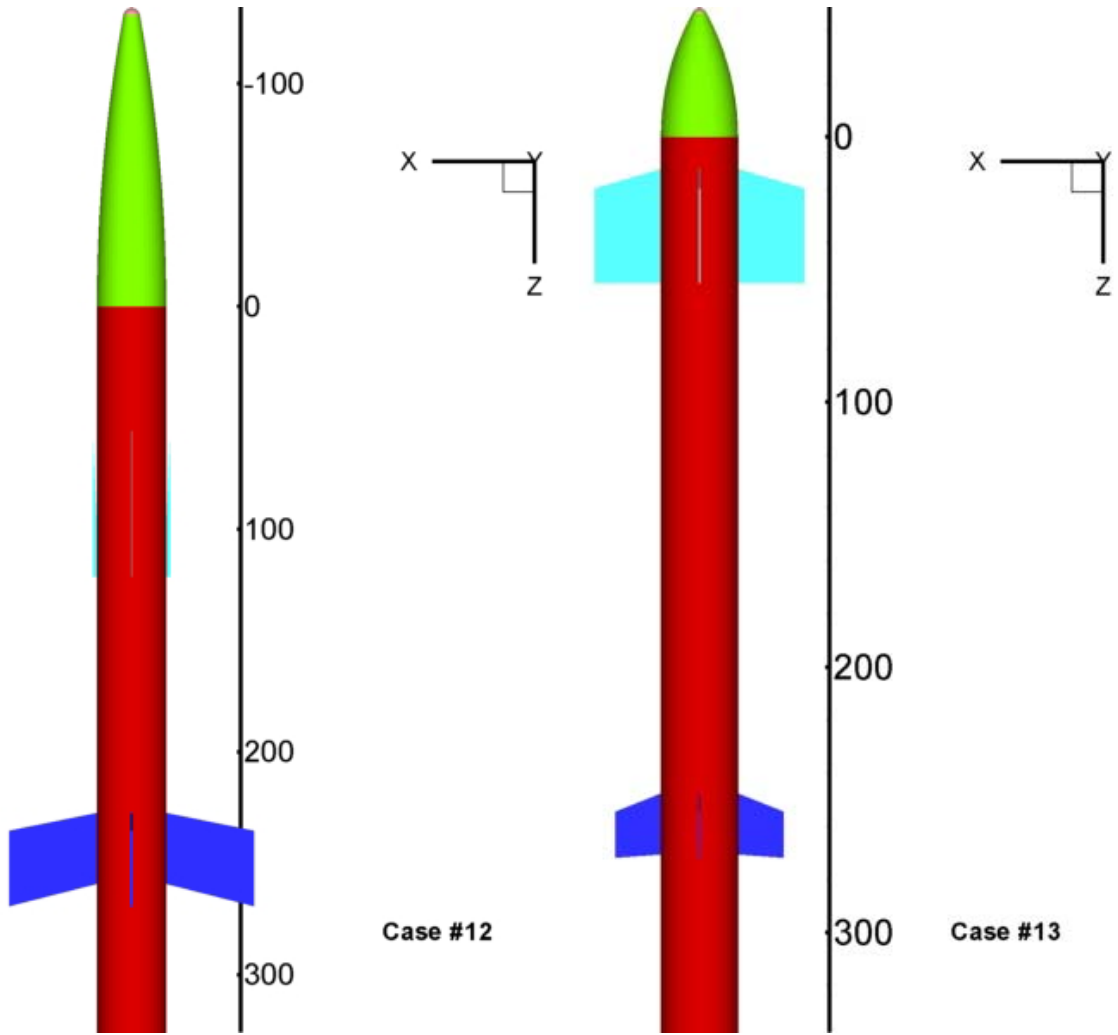


Figure 11: External View of Best Matches (cont.)

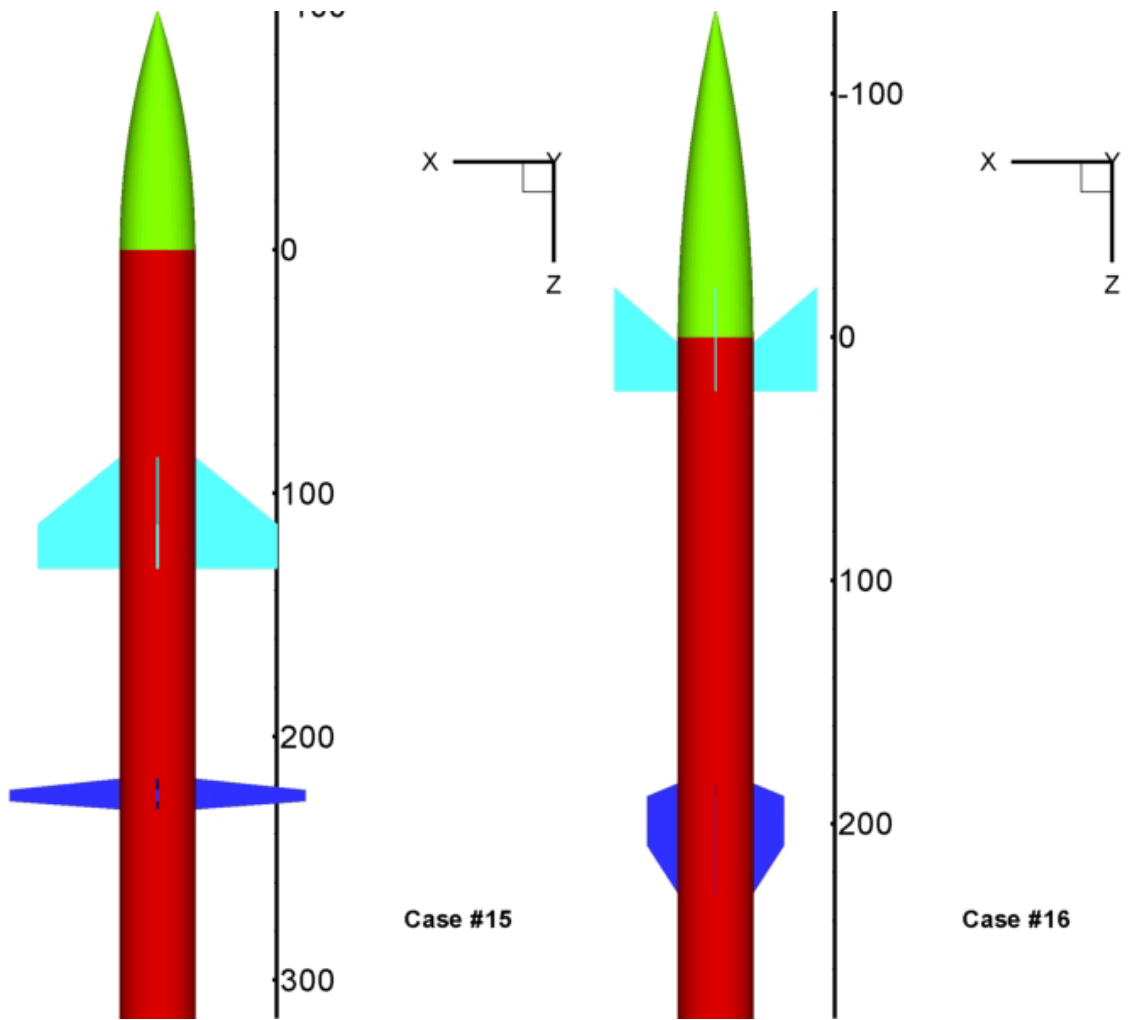


Figure 11: External View of Best Matches (cont.)

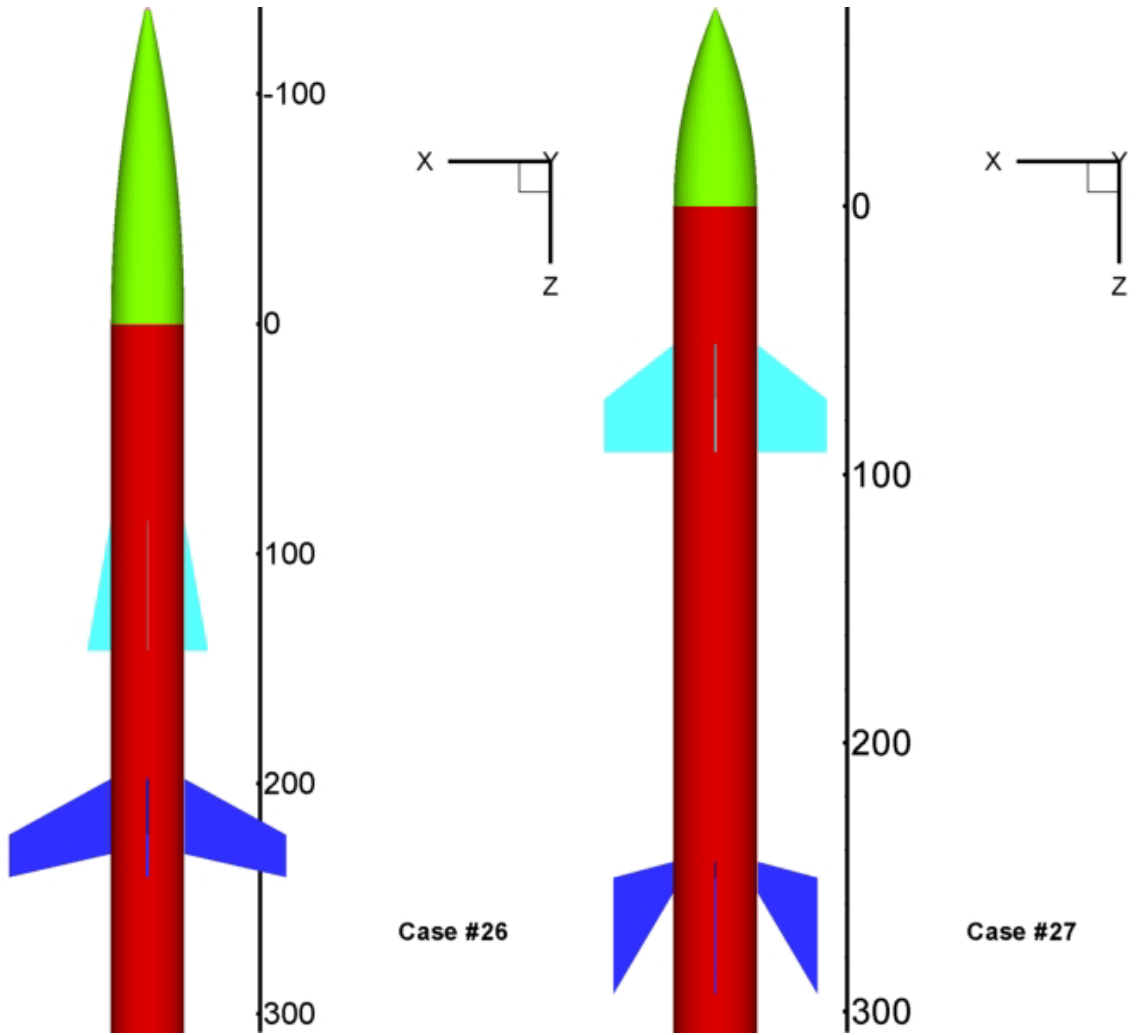


Figure 11: External View of Best Matches (cont.)

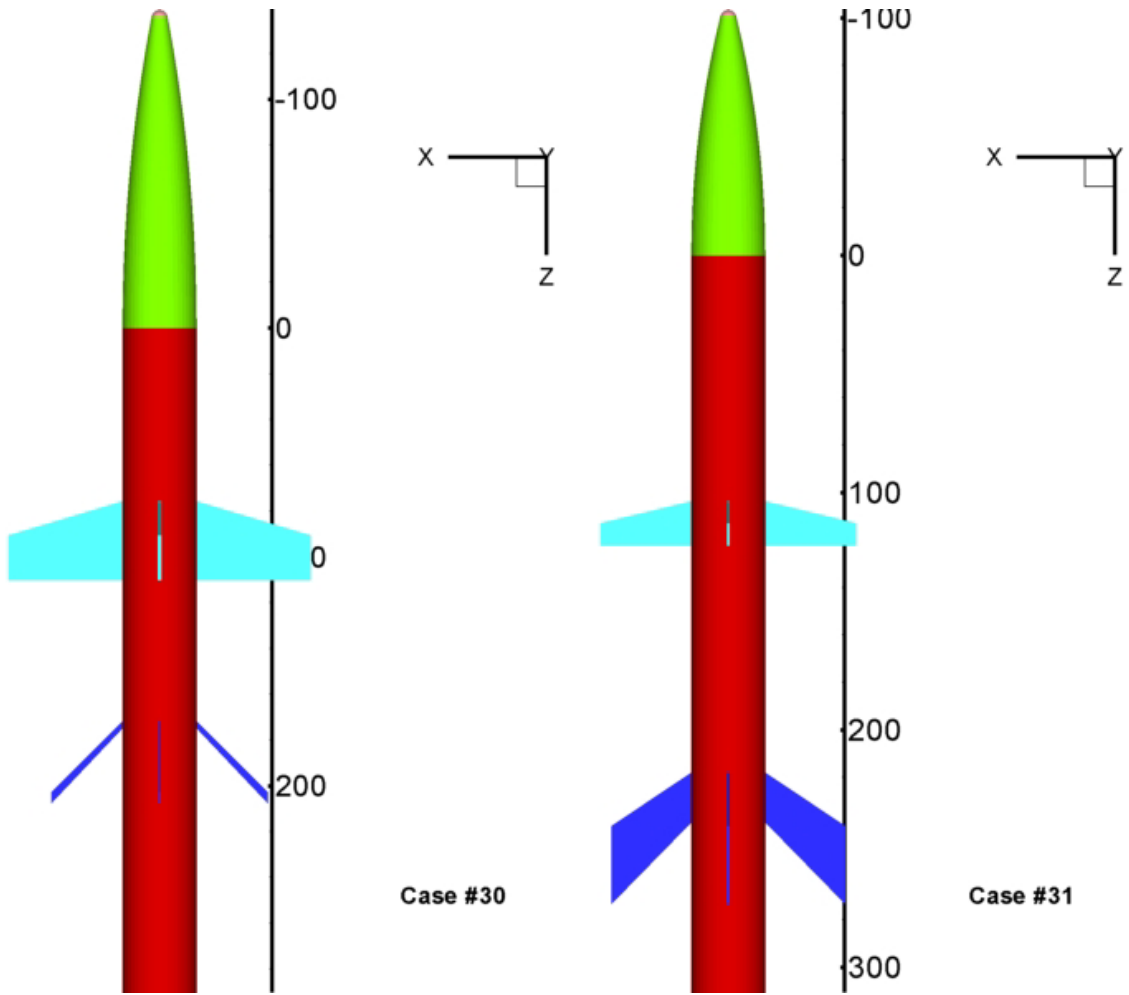


Figure 11: External View of Best Matches (cont.)



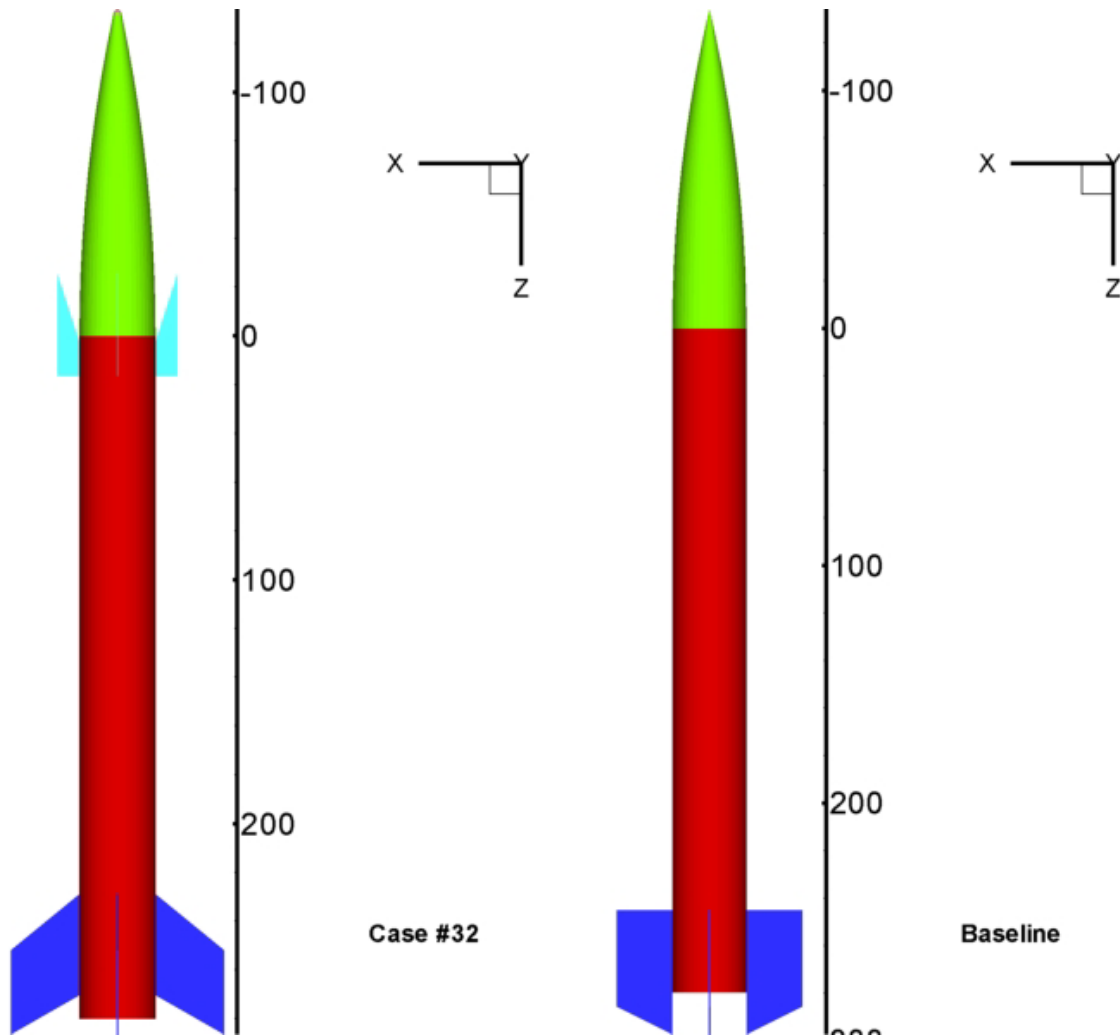


Figure 11: External View of Best Matches (cont.)

As mentioned elsewhere, the target values for the four matching goals are effectively known information about the baseline missile. It is pertinent to examine whether the findings of this investigation are diminished when fewer goals are used, although a thorough examination of variable goal sets would significantly increase the required computation time and complexity of analysis and is beyond the scope of this thesis. Thus, only the limited case of dropping the initial mass goal will be presented

here. The target value to match for this goal would typically be the most difficult information to procure, among the four goals used in the primary investigation, so it is important to know whether the reverse-engineering process can still be successful without initial mass data for the baseline.

Three cases were executed without the mass goal: Case 1, which has no locked parameters, Case 3, which has a locked diameter, and Case 32, which has all five studied parameters locked. The interim case with only diameter locked was chosen because it was expected that diameter would play a more significant role in the absence of initial mass data, and in fact that may be why diameter is not shown to be a vital parameter in the primary investigation. Figures 12-15 show that time of flight, range, and altitude are reasonably matched in these cases with or without the mass goal; of course, range and altitude are themselves still goals, and time of flight is generally proportional to altitude in a ballistic flight. The thrust curves and mass curves are more stratified and provide more insight into the effect of eliminating the mass goal.

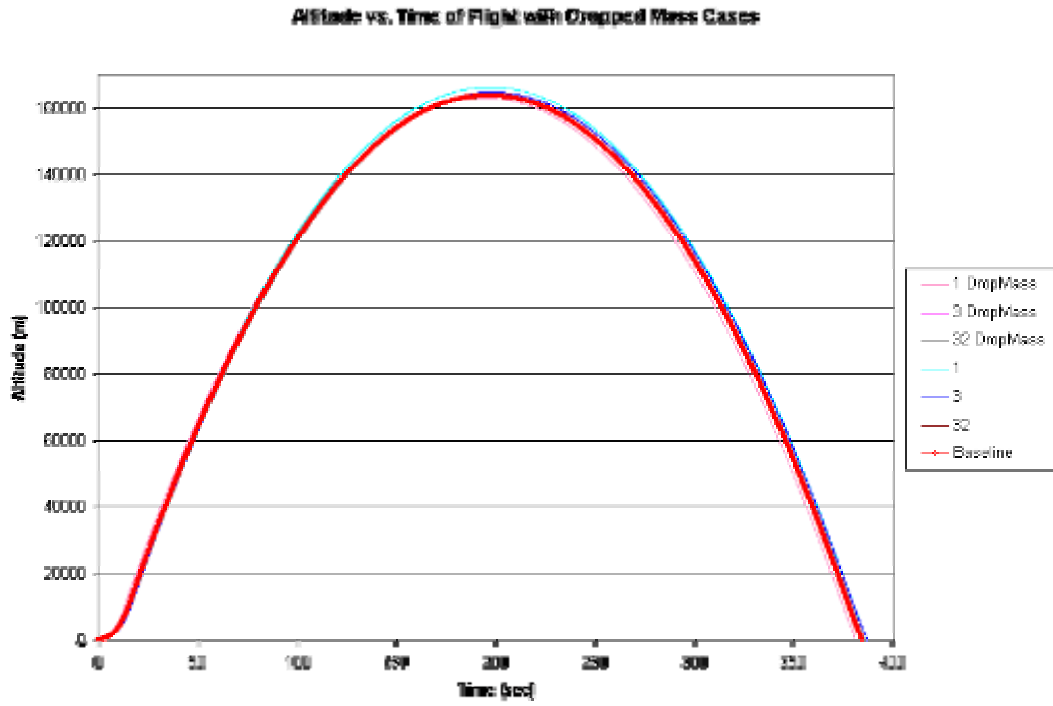


Figure 12: Altitude vs. Time of Flight with Dropped Mass Cases

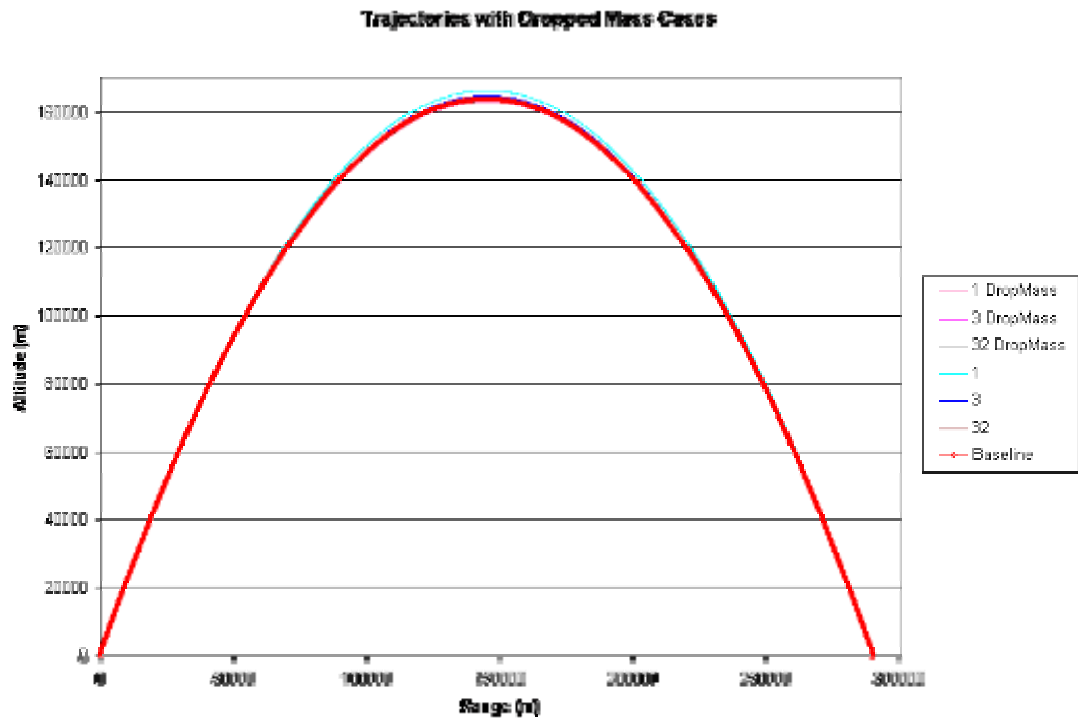


Figure 13: Trajectories with Dropped Mass Cases

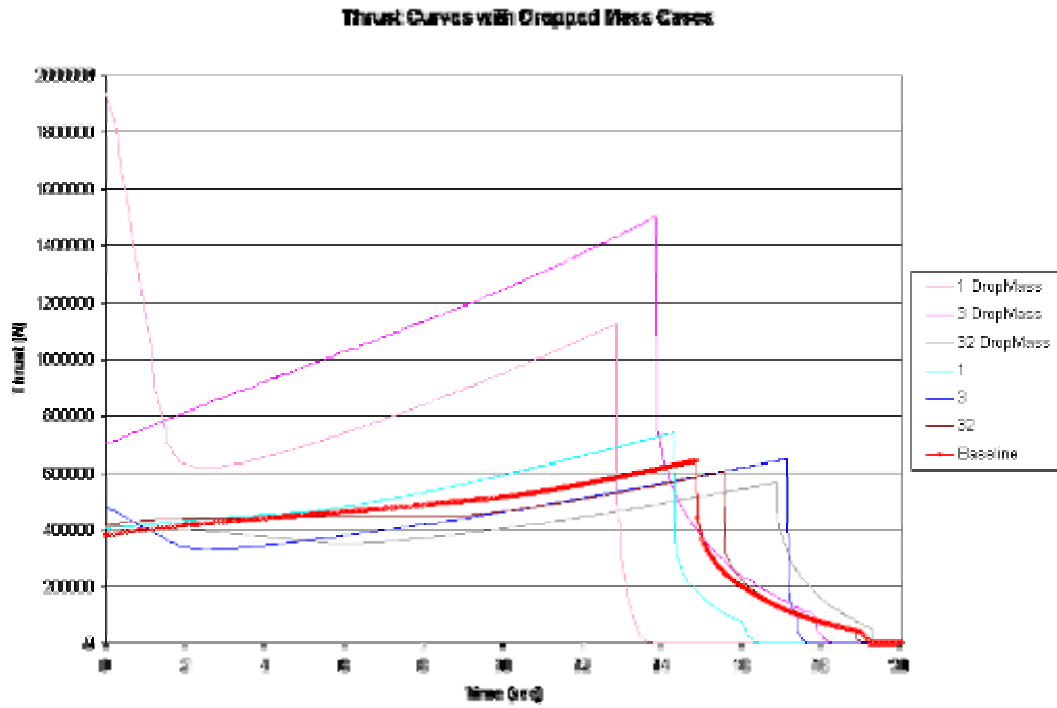


Figure 14: Thrust Curves with Dropped Mass Cases

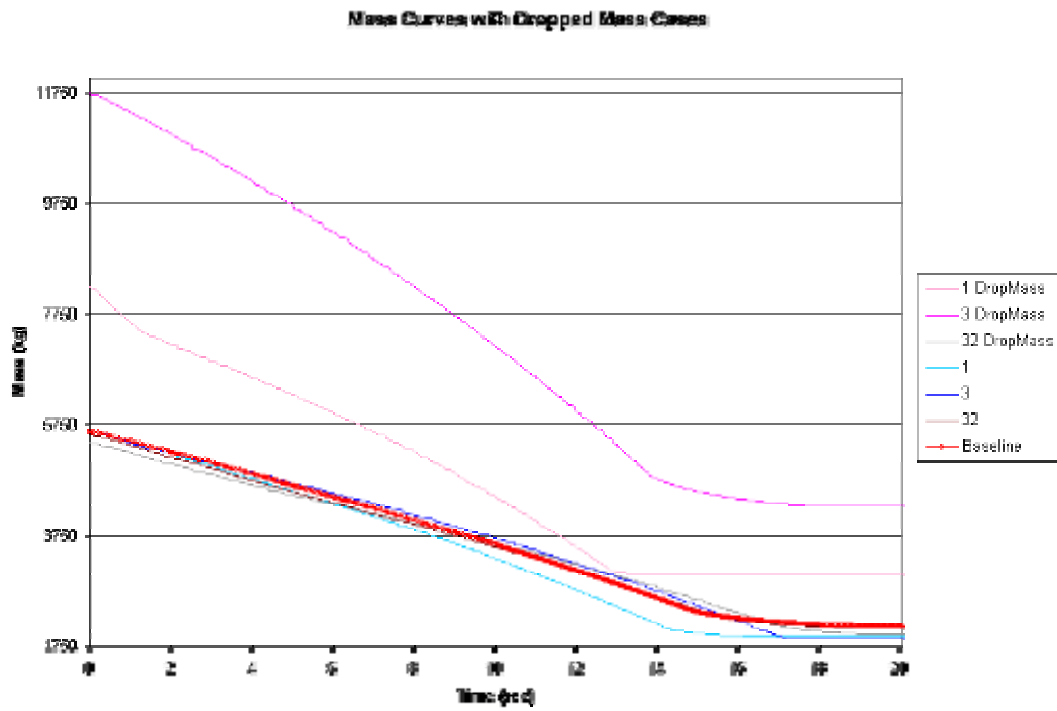


Figure 15: Mass Curves with Dropped Mass Cases

Regarding the thrust curves and mass curves for the dropped mass cases, the significant finding is that the lack of initial mass target data can be overcome with enough locked parameters. Case 32 is an excellent match with or without the mass goal, with the full four-goal analysis having just a slight edge in accuracy. But Cases 1 and 3 show that lacking the mass goal can be a serious disadvantage in the absence of locked parameters. These limited results do not show the predicted effect of locking the diameter value when mass is not a goal. Analysis of additional cases could possibly expose a co-dependence of diameter with fineness ratio or some other parameter which would indeed overcome the lack of mass goal data.

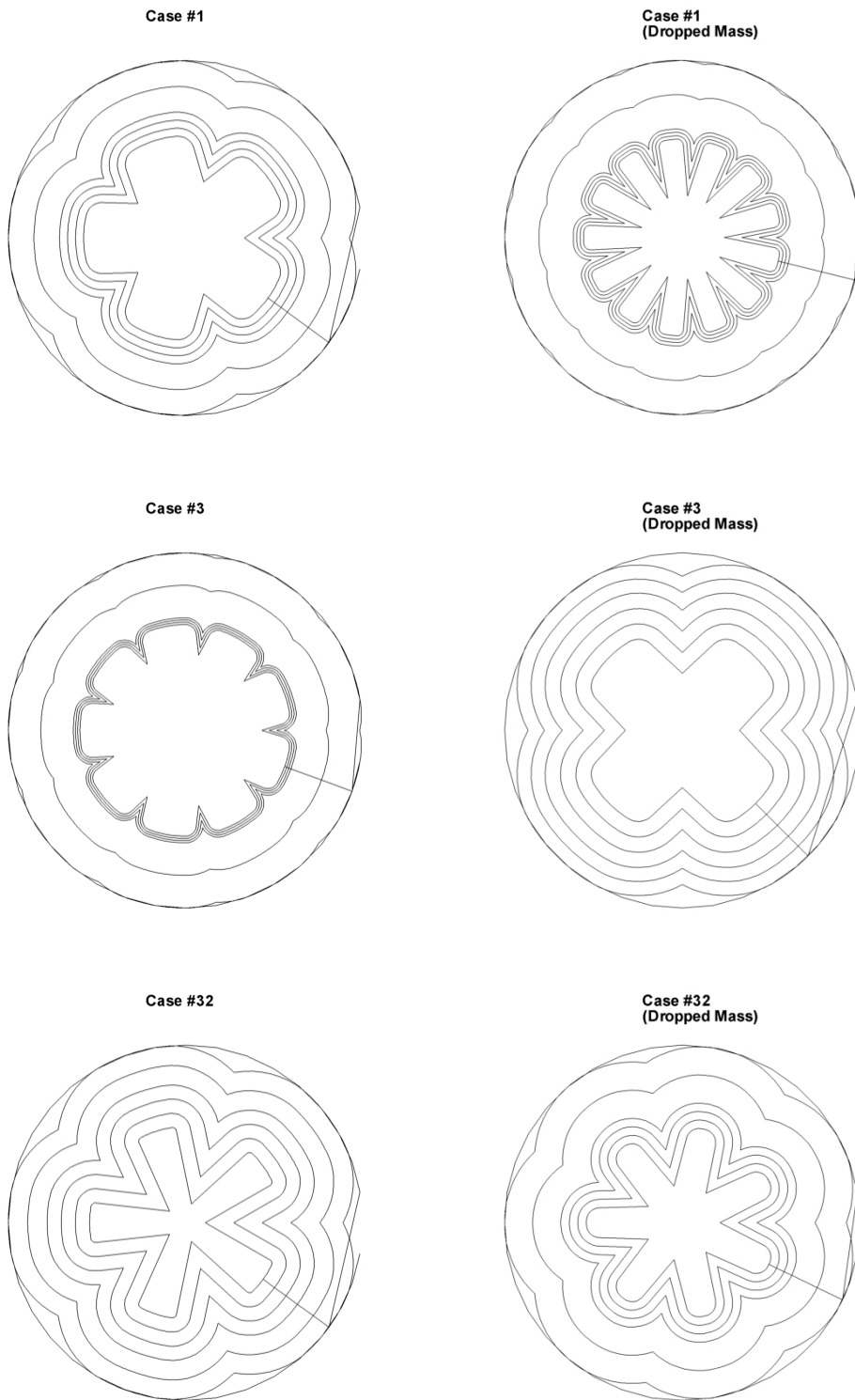


Figure 16: Propellant Grain Cross Sections with Dropped Mass Cases

Figure 16 compares the grain cross sections of the original cases and dropped mass cases. Case 1 is a somewhat poorer match without the mass goal, while Case 3 is somewhat better than the original case, but none of these are close matches. Case 32 again shows that, with enough locked parameters, the GA can find a good match even without the initial mass goal. The number of star points is off by two, but otherwise the grain cross-section is a good match for the baseline. The original case with all four goals is clearly superior, though.

Figure 17 shows the external geometry of the dropped mass cases. As in the primary investigation, the fin configurations are erratic due to the aerodynamic nature of ballistic missile flights.

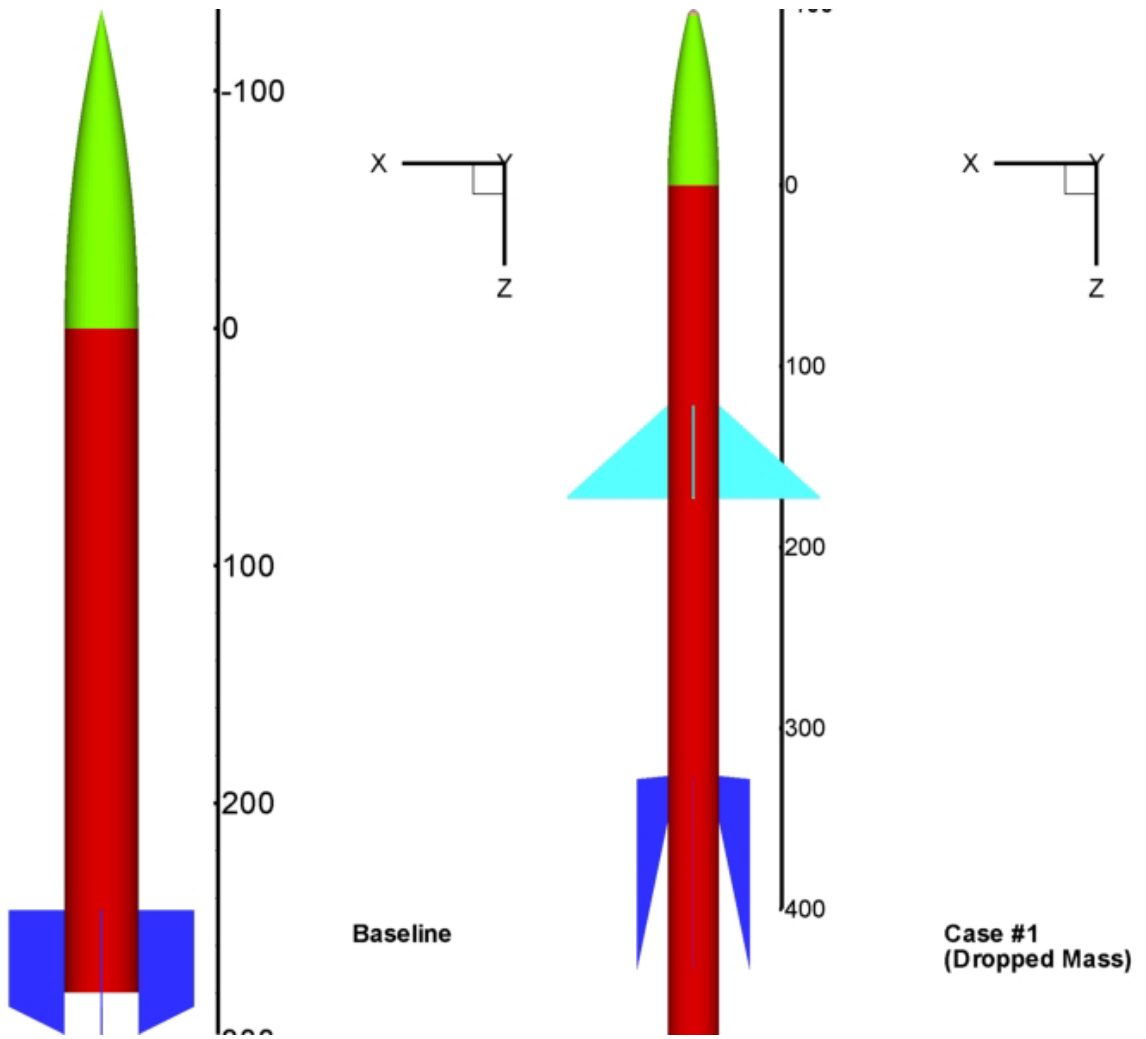


Figure 17: External View of Dropped Mass Cases



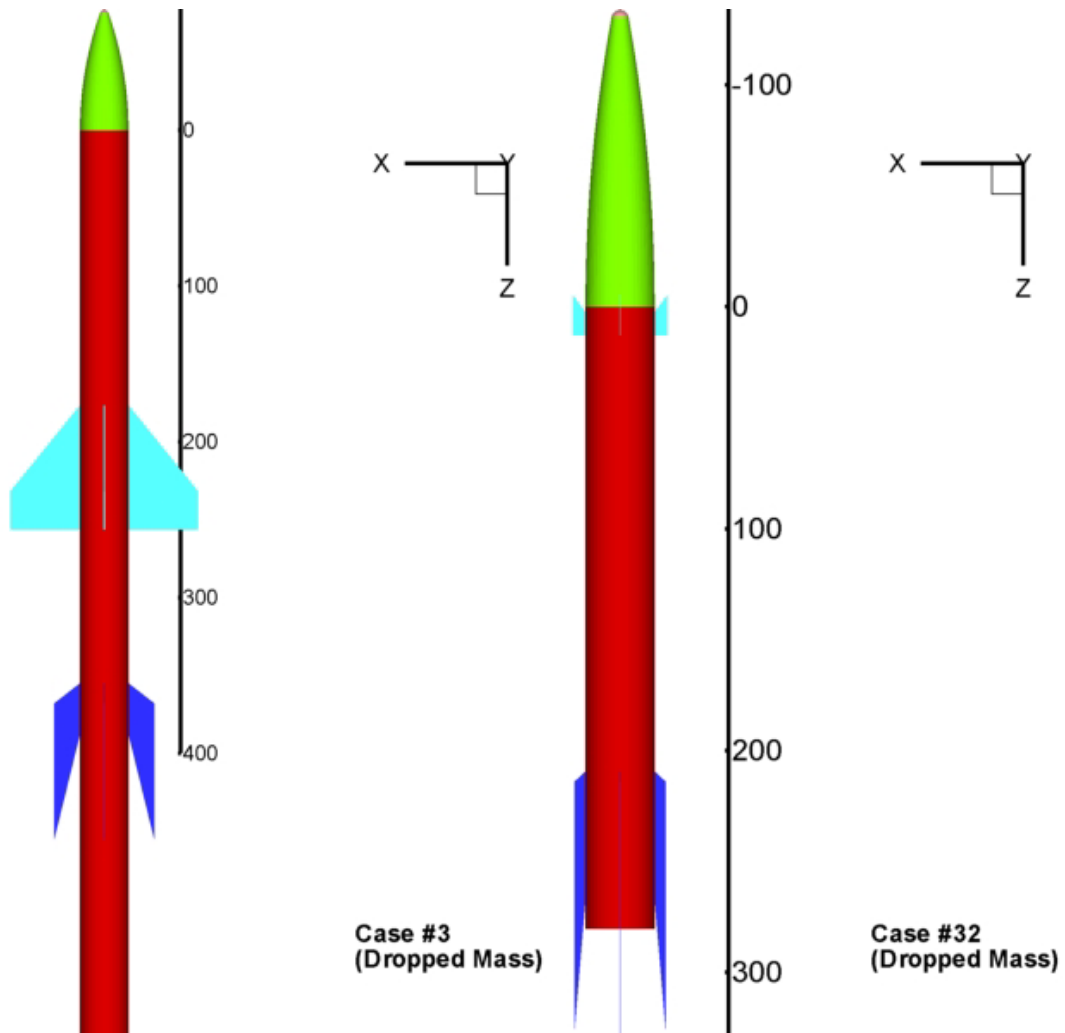


Figure 17: External View of Dropped Mass Cases (cont.)

### Concluding Comments

The frequency of propellant radius among the best matching cases indicates that some knowledge of the internal grain geometry is a significant help in finding the best reverse-engineering matches for a solid rocket propelled missile. However, fineness ratio is nearly as important, and this parameter is far more likely to be available from even

limited surveillance data, since it can be extrapolated from a photograph regardless of scale. Even more promising is that the GA is capable of finding a quality reverse-engineering match even with no locked design parameters, though sufficient performance data for the fitness goals are still required. The value of executing multiple cases through the GA has also been shown, particularly when comparing the best thrust curve profiles to determine characteristics of the propellant grain. In the reverse-engineering analysis, the importance of inspecting populations by hand to determine the true best performer has also been shown. Finally, the effect of changing the goal set has been shown as significant to the reverse-engineering process, although locking enough design parameters may be sufficient to overcome a set of fewer matching goals.

### Recommendations

Further investigation of this topic could include additional design parameters in the test matrix, as well as more exhaustively studying the importance of performance characteristics in the fitness goals. Considerably more computation time could be devoted to analyzing these cases through additional generations and/or increased population size, to examine whether the trends endure even as the random elements of the GA are diffused through sheer quantity of simulations. The results of this thesis could be validated by applying the same principles to additional baseline missiles which may have completely different scales and grain geometries. With a more advanced model, these methods could also be tested on missiles with multiple stages and missiles with propellant grains which are not uniform length-wise, such as tapered or composite grains.

## REFERENCES

1. "Genetic Algorithms for Missile Analysis," J.E. Burkhalter, R.M. Jenkins, and R.J. Hartfield, Final Report, Missile and Space Intelligence Center, Redstone Arsenal, AL Feb, 2003
2. Hartfield, Roy J., Burkhalter, John E., Jenkins, Rhonald M., and Metts, Jonathan, "Genetic Algorithm Upgrade FINAL REPORT", submitted to Missile and Space Intelligence Center, Redstone Arsenal, Alabama 35898, 10 June 2005, Reference Contract No. HHM402-04-P-0061.
3. Wollam, J., Kramer, S., Campbell, S., "Reverse Engineering of Foreign Missiles via Genetic Algorithm", AIAA Paper 2000-0685, 38<sup>th</sup> Aerospace Sciences Meeting & Exhibit, Reno, NV, January 2000.
4. Anderson, M. B., "Users Manual for IMPROVE Version 2.8: An Optimization Software Package Based on Genetic Algorithms", Sverdrup Technology Inc. / TEAS Group, Eglin AFB, Florida, March 6, 2001.

5. Yang, C. D., Luo, C. C., Liu, S. J., and Chang, Y. H., "Applications of Genetic-Teguchi Algorithm in Flight Control Designs", *Journal of Aerospace Engineering*, Vol. 18, No. 4, October 2005, pp. 232-241
6. Sforzini, R. H., Foster, W. A., and Johnson, J. S., "A Monte Carlo Investigation of Thrust Imbalance of Solid Rocket Motor Pairs", submitted to Marshall Space Flight Center, AL 35812, November 1974, NASA-CR-120700.
7. Perhinschi, M.,G., "A Modified Genetic Algorithm for the Design of Autonomous Helicopter Control System," AIAA-97-3630, Presented at the AIAA Guidance, Navigation, and Control Conference, New Orleans, LA, August 1997.
8. Karr, C.L., Freeman, L.M., and Meredith, D.L., "Genetic Algorithm Based Fuzzy Control of Spacecraft Autonomous Rendezvous," NASA Marshall Space Flight Center, Fifth Conference on Artificial Intelligence for Space Applications, 1990.
9. Mondoloni, S., "A Genetic Algorithm for Determining Optimal Flight Trajectories", AIAA Paper 98-4476, AIAA Guidance, Navigation, and Control Conference and Exhibit, August 1998.
10. Torella, G., Blasi, L., "The Optimization of Gas Turbine Engine Design by Genetic Algorithms", AIAA Paper 2000-3710, 36<sup>th</sup> AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, July 2000.

11. Jang, M., and Lee, J., "Genetic Algorithm Based Design of Transonic Airfoils Using Euler Equations", AIAA Paper 2000-1584, Presented at the 41<sup>st</sup> AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, April 2000.
12. Jenkins, Rhonald M., Hartfield, Roy J., and Burkhalter, John E., "Optimizing a Solid Rocket Motor Boosted Ramjet Powered Missile Using a Genetic Algorithm", AIAA 2005-3507 presented at the Forty First AIAA/ASME/SAE/ASEE Joint Propulsion Conference, Tucson, AZ, July 10-13, 2005.
13. Anderson, M.B., Burkhalter, J.E., and Jenkins, R.M., "Design of an Air to Air Interceptor Using Genetic Algorithms", AIAA Paper 99-4081, presented at the 1999 AIAA Guidance, Navigation, and Control Conference, Portland, OR, August 1999.
14. Perez, R.E., Chung, J., Behdinan, K., "Aircraft Conceptual Design Using Genetic Algorithms", AIAA Paper 2000-4938, Presented at the 8<sup>th</sup> AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, September 2000.
15. Schoonover, P.L., Crossley, W.A., and Heister, S.D., "Application of Genetic Algorithms to the Optimization of Hybrid Rockets", AIAA Paper 98-3349, 34<sup>th</sup> AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, July 1998.

16. Burkhalter, J., Jenkins, R., Hartfield, R., Foster, W., Witt, J., and Heiser, M., "Missile Design Systems Developed With Genetic Algorithms," Final Report for Dynetics, Inc., P.O. Box 5500, Huntsville, AL 35814 and U.S. Army Aviation & Missile Command, Redstone Arsenal, Alabama 35898, October 31, 2001, Prime Contract No. DAAH01-96-C-R194/P39393.
  
17. Billheimer, J.S., "Optimization and Design Simulation in Solid Rocket Design," AIAA Paper 68-488, Presented at the 3<sup>rd</sup> AIAA Solid Propulsion Conference, June, 1968.
  
18. Anderson, M.B., Burkhalter, J.E., and Jenkins, R.M., "Multi-Disciplinary Intelligent Systems Approach to Solid Rocket Motor Design, Part I: Single and Dual Goal Optimization", AIAA Paper 2001-3599, presented at the 37<sup>th</sup> AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit, Salt Lake City, UT, July 2001.

## APPENDIX A: AGARAT Manual

This investigation makes use of a custom-built program called the Advanced Genetic Algorithm Results Analysis Tool, or AGARAT. It consists of two pieces: a Fortran-coded shell program that automates single-run simulations in the solid rocket modeling suite, and a package of Microsoft Excel ® macros coded in Visual Basic for Applications (VBA) which quickly generate useful plots that aid the user in comparing performance results among a set of members. AGARAT conveniently provides a more thorough look at several of the best members (not just one top-sorted member) of a generation, typically the final generation.

For input, AGARAT requires the materials properties file (boxsave.dat) and any population file from the IMPROVE genetic algorithm (GA). The original GA input file, Gannl.dat, is not required. The user is prompted to type in the population data filename (Ex. "population.00050" or "looperout32.dat") and the number of members to be analyzed. AGARAT assumes that minimization sorting is in effect and searches the population file to find the chosen number of members, starting with the GA's top-sorted member. The program then converts the parameter data for these members from horizontal to vertical format and saves the parameters for each one in a uniquely named file, starting with Snglerun.1. AGARAT then calls the main suite of missile simulation codes for each unique parameter file. Between each call, the important output files for

the analysis just completed are stored under corresponding unique filenames, such as grainplot.1, rangedata.1, and geomout\_matlab.1. This process continues until all of the members have been analyzed and AGARAT has stored all the output files. Please note that these files will be overwritten when AGARAT is executed the next time, so copy important output files to a separate directory.

The Excel ® macros are found in the AGARAT\_Macros.xls workbook. There are macros set up to analyze five, ten, or fifteen members at a time, but it is a simple matter to customize the VBA codes for additional or interim numbers of members. Included macros generate plots for trajectory, thrust curves, etc. with each member represented as a separate data series on the same plot. Solid rocket grain cross-sections are drawn into a separate chart for each member. In a reverse-engineering application, a series for the baseline data can also be plotted. From these plots, comparisons may be observed which are completely independent of the objective function used by the GA to sort the members. Thus, these plots can aid in choosing top performers in categories such as thrust profiles and grain cross-sections which may be difficult to program into fitness goals in the GA. After the macros have been executed, the plots may be customized freely for better use in a paper or presentation. In order to maintain the function of the macros, the user must save the workbook under a different name or close the workbook without saving changes. If this rule is not followed, macro functionality in AGARAT\_Macros.xls can be restored by deleting all non-default worksheets and all plots and other data and then saving the workbook to its original condition, in which it is completely blank (except for the macros).