

**A HYBRID PROBABILISTIC METHOD TO ESTIMATE DESIGN
MARGIN**

A Thesis
Presented to
The Academic Faculty

by

Bradford E. Robertson

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Aerospace Engineering

Georgia Institute of Technology
December 2013

Copyright © 2013 by Bradford E. Robertson

A HYBRID PROBABILISTIC METHOD TO ESTIMATE DESIGN MARGIN

Approved by:

Prof. Dimitri Mavris, Committee Chair
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Jeffrey Zweber
Air Force Research Laboratory

Prof. Daniel Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Dr. Neil Weston
School of Aerospace Engineering
Georgia Institute of Technology

Prof. Brian German
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: November, 15th 2013

Cleared for Public Release. Air Force Case Number: 88ABW-2013-4331

Ad Majorem Dei Gloriam

ACKNOWLEDGEMENTS

As I conclude my journey through graduate school, I would like to take time to acknowledge those who helped make this experience possible and fulfilling.

First and foremost, I would like to express my gratitude towards my advisor, Dr. Dimitri Mavris. Under his tutelage, I have learned innumerable lessons about both life and engineering over the past several years. I will always be grateful for the opportunities that he gave me.

I would also like to thank Dr. Jeffrey Zweber and Barry Hellman from AFRL for sponsoring this and related work into reusable launch vehicles.

Finally, I would like to thank my parents for their support of my education. Thanks for standing by me when I switching into engineering halfway through my undergraduate career. I think it ended up working out.

TABLE OF CONTENTS

DEDICATION	iv
ACKNOWLEDGEMENTS	v
LIST OF TABLES	xi
LIST OF FIGURES	xii
SUMMARY	xvii
I INTRODUCTION	1
1.1 Launch Vehicles	1
1.1.1 Current Launch Vehicles	1
1.1.2 The Physics of Space Launch	3
1.2 Uncertainty in Space and Launch Vehicles Performance Prediction	13
1.2.1 Mass Growth in Space and Launch Vehicles	13
1.2.2 Propulsion	20
1.2.3 The Use of Margin	21
1.2.4 Observations	25
II UNCERTAINTIES IN SPACE AND LAUNCH VEHICLES	28
2.1 Endogenous and Exogenous Sources of Uncertainty	29
2.1.1 Endogenous Sources of Uncertainty	29
2.1.2 Exogenous Sources of Uncertainty	30
2.1.3 The Effects of Both Endogenous and Exogenous Uncertainties	31
2.2 Treatment of Uncertainties in Literature	31
2.2.1 Ecology and Conservation Biology	33
2.2.2 Civil Engineering	36
2.2.3 Structural Engineering	37
2.2.4 Systems Engineering	39
2.2.5 Modeling and Simulation	39
2.2.6 Space Architectures and Systems	40
2.2.7 Design of Complex Systems	40
2.3 Taxonomy of Uncertainty	43

2.3.1	Endogenous Uncertainties	45
2.3.2	Exogenous Uncertainties	47
2.4	Uncertainty Elicitation	50
2.4.1	Traditional Engineering Analyses	50
2.4.2	Programmatic Risk and Opportunity Analyses	51
2.4.3	Technical Risk and Opportunity Analyses	51
2.4.4	Observation	52
III	METHODS OF ESTIMATING DESIGN MARGIN	53
3.1	Principles for Design Margin Estimating Techniques of a Novel Concept	54
3.1.1	AACE Recommended Practice	54
3.1.2	Principles Necessary for Novel Concept Margin Estimation	56
3.2	Predetermined Percentage	59
3.2.1	Description	59
3.2.2	Evaluation	59
3.3	Historical Regression	61
3.3.1	Description	61
3.3.2	Evaluation	64
3.4	Expert Opinion	65
3.4.1	Description	65
3.4.2	Evaluation	65
3.5	Probabilistic Techniques	67
3.5.1	Description	67
3.5.2	Evaluation	74
3.6	Gaps	76
IV	EXTENDING PROBABILISTIC ESTIMATION TECHNIQUES	79
4.1	Application of Range Estimating to Potential Baseline Changes	79
4.1.1	Space Shuttle Combinatorial Study	79
4.2	A Hybrid Methodology	85
4.2.1	Methodology Development	86
4.2.2	Adapting Scenario-Based Forecasting to Vehicle Design	90

V	EXECUTABLE MORPHOLOGICAL ANALYSIS DATA STRUCTURES AND ALGORITHMS	96
5.1	Previous Work into Extending Morphological Analysis	96
5.2	Generalized Executable Morphological Analysis	97
5.2.1	Data Structures of Executable Morphological Analysis	98
5.3	Specific Realization of EMA Data Structures	102
5.3.1	Required Information in the Executable Matrix of Alternatives	103
5.3.2	Relationships	108
5.4	Specific Realization of EMA Algorithms	113
5.4.1	Selecting a Combination	114
5.4.2	Potential Options	123
5.4.3	Direct Calculation of PMF	125
5.4.4	Experiment 1– Monte Carlo Simulation of EMA	127
5.5	Implementation of EMA Data Structures	148
5.5.1	Object Oriented Formulation	148
5.5.2	Running EMA	156
5.5.3	Specific Enablers of EMA	158
VI	METHODOLOGY DEVELOPMENT	160
6.1	Problem Setup	160
6.2	Forecasting Utilizing Executable Morphological Analysis	162
6.2.1	The Identification of Parameters and Conditions	162
6.2.2	Identification of Condition Attributes	164
6.2.3	Identification of Relationships	166
6.2.4	Analysis	170
6.3	Combined Margin Analysis	173
VII	SPACE SHUTTLE ORBITER STUDY	175
7.1	Space Shuttle Orbiter Design Changes	175
7.1.1	ATP Orbiter	175
7.1.2	Program Readiness Review Orbiter	176
7.1.3	150K Orbiter	179
7.1.4	Vehicle 3/4 Orbiter	179

7.1.5	Vehicle 5/6 Orbiter	181
7.2	Analyzing the Space Shuttle Orbiter	181
7.2.1	Baseline-Specific Growth	184
7.2.2	EMA	184
7.3	Results and Conclusions	190
VIII FAST PROGRAM PROOF OF CONCEPT		193
8.1	Future-Responsive Access to Space Technologies Program	193
8.2	Range Estimating	194
8.2.1	Work Breakdown Structure	194
8.2.2	Estimate Ranges for Individual Subsystems	195
8.2.3	Identify Dependencies	197
8.2.4	Simulation	197
8.3	EMA	199
8.3.1	Identification of Model Parameters	199
8.3.2	Identification of Conditions	203
8.3.3	Identification of Condition Attributes	206
8.3.4	Identification of Relationships	211
8.3.5	Analysis	221
8.4	Margin Selection	229
8.5	Additional Use Cases for EMA	232
8.5.1	EMA Model Validation	233
8.5.2	Subsystem Sensitivity	236
8.5.3	Subsystem Design Decisions	239
8.5.4	Subsystem Risk Analysis	243
8.5.5	Requirements Risk Mitigation	246
IX CONCLUSIONS AND FUTURE WORK		253
9.1	Conclusions	253
9.2	Future Work	258
9.3	Contributions	259
APPENDIX A — CODE FOR OBJECT-ORIENTED IMPLEMENTATION		261

REFERENCES	314
VITA	324

LIST OF TABLES

1	Specific Orbital Energies of Common Orbits	6
2	Typical Values of I_{sp} [143]	8
3	Launch Vehicle Historical Mass Growth [23]	14
4	Manned Spacecraft Historical Mass Growth [54]	14
5	Advanced Concept Mass Growth [130]	17
6	Rocket Engine Historical Mass Growth [144]	17
7	AIAA Mass Change Codes and Definitions [6]	18
8	Taxonomy of Uncertainties for Space Architectures and Systems [142]	41
9	Linkages Between Major Changes and Substructures	82
10	Joint Probability Distribution for Example Matrix with Probabilities	107
11	Ratios of Probabilities for Figure 32	109
12	Ratios of Probabilities for Figure 34	110
13	Ratios of Probabilities for Figure 36	111
14	Orbiter Dry-Weight Predictions	190
15	Comparisons Between Flight-Weight Orbiters and Predicted Weights	192
16	FAST RFS F Weight Breakdown Structure	196
17	Percentage Offsets for Range Estimating	197
18	Organization of Parameters by Subsystem	201
19	Subsystem Margin Estimation from EMA	223
20	Subsystem Margin Correlations	227
21	Total Margin Estimation	229
22	Total Margin Estimation via Subsystems	230
23	Total Margin Estimation	232
24	Total Margin Estimation via Subsystems	232

LIST OF FIGURES

1	U.S. Air Force Reusable Booster Concept [51]	4
2	Atlas V 521 Launch Profile [134]	12
3	Dry Mass Growth of Satellites [53]	15
4	Dry Mass Growth of Space Missions [16]	16
5	Margin and Weight Growth vs. Time During a Development Program [6] .	22
6	System Commitment, Ease of Change, System-Specific Knowledge, and Cost Incurred vs. Time During a Development Program [17]	24
7	Dry Mass Growth of Skylab [54]	32
8	Taxonomy of Uncertainties in Ecology [108]	33
9	Taxonomy of Uncertainties in Civil Engineering [7]	36
10	Taxonomy of Uncertainties in Structural Engineering [84] adapted from [132]	37
11	Taxonomy of Uncertainties in Modeling and Simulation [99]	40
12	Taxonomy of Uncertainties in the Design of Complex Systems [132]	42
13	Taxonomy of Uncertainties in the Development of Space and Launch Vehicles	44
14	Principles of Estimation Techniques vs. Standard Techniques [1]	57
15	Jet Propulsion Laboratory Mass Growth Standard [150]	60
16	Spacecraft Mass versus Man Days in Space [54]	62
17	Vertical Tail Mass vs. Tail Area with Bounds of 1 Standard Deviation [24] .	63
18	Double Triangular Distribution [2]	70
19	Morphological Matrix of Historical Changes to the Orbiter	81
20	Space Shuttle Launch Loads DAG	83
21	Space Shuttle Gliding Loads DAG	83
22	Space Shuttle Landing Loads DAG	84
23	Sample Scenario Matrix	88
24	Sample Morphological Field	92
25	Sample Cross Consistency Matrix	93
26	Notional Matrix of Alternatives	94
27	A Traditional Matrix of Alternatives	98
28	A Notional Executable Matrix of Alternatives	99

29	A Notional Relationship Matrix	100
30	One-Way Relationship Matrix	102
31	Example Matrix with Probabilities	107
32	Expanded Example Matrix with Probabilities	109
33	Expanded Example Matrix After Selection of Condition A2	110
34	Expanded Example Matrix After Normalization of Parameter B	110
35	Expanded Example Matrix After Selection of Condition A2	111
36	Expanded Example Matrix After Normalization of Parameter B	111
37	Executable Matrix of Alternatives used for Selection Example	115
38	Relationship Matrix used for Selection Example	116
39	The Selection of Condition A1 and the Activation of Relationships	118
40	The Selection of Condition A1 and the Normalization of Probabilities	119
41	The State of the Executable Matrix of Alternatives after the selections of A1, B4, and $\Gamma 4$	120
42	The State of the Executable Matrix of Alternatives after the selections of A1, B4, $\Gamma 4$, and $\Delta 1$	121
43	The State of the Executable Matrix of Alternatives after the selections of A1, B4, $\Gamma 4$, $\Delta 1$, and E1	122
44	Two Different Orderings of Selecting A2 and B2 from Figure 32	126
45	Monte Carlo Procedure for EMA Models	129
46	The directly computed CDF for the small example matrix without relationships	131
47	The CDF for the small example matrix without relationships computed via Monte Carlo simulation	132
48	Expected value and confidence interval versus number of Monte Carlo simu- lations for the small example matrix without relationships	133
49	80% quantile value and confidence interval versus number of Monte Carlo simulations for the small example matrix without relationships	134
50	The directly calculated CDF for the example matrix with relationships	135
51	The CDF for the small example matrix with relationships calculated via Monte Carlo simulation	136
52	Expected value and confidence interval versus number of Monte Carlo simu- lations for the small example matrix with relationships	137
53	80% quantile value and confidence interval versus number of Monte Carlo simulations for the small example matrix with relationships	138

54	Medium-sized executable matrix of alternatives	139
55	The CDF for a medium-sized morphological field with relationships calculated via Monte Carlo simulation for the medium-sized executable matrix of alternatives	140
56	Expected value value and confidence interval versus number of Monte Carlo simulations for the medium-sized executable matrix of alternatives	141
57	80% quantile value and confidence interval versus number of Monte Carlo simulations for the medium-sized executable matrix of alternatives	142
58	Large executable matrix of alternatives	144
59	The CDF for a large morphological field relationships calculated via Monte Carlo simulation	145
60	Expected value value and confidence interval versus number of Monte Carlo simulations for the large executable matrix of alternatives	146
61	80% quantile value and confidence interval versus number of Monte Carlo simulations for the large executable matrix of alternatives	147
62	Generic Object-Oriented Implementation	152
63	Specific Object-Oriented Implementation for a Weight Forecasting Problem	155
64	Hybrid Forecasting Method	161
65	A Two-Level Executable Matrix of Alternatives	168
66	The Space Shuttle Orbiter at Authorization to Proceed [61]	177
67	The Space Shuttle Orbiter at its Program Readiness Review [61]	178
68	The 150k Orbiter [61]	180
69	The Vehicle 3/4 Orbiter Configuration [61]	182
70	The Final Space Shuttle Orbiter Configuration [61]	183
71	Space Shuttle Orbiter Design Change Morphological Field	186
72	Space Shuttle Mass Changes as of July 1973[95]	187
73	Space Shuttle Orbiter Design Change Executable Matrix of Alternatives	189
74	CDF of Projects Changes to Orbiter Weight	191
75	FAST RFS F [12]	194
76	CDF of Total Vehicle Weight from Range Estimating	198
77	CDF of Total Margin from Range Estimating	198
78	Morphological Field for FAST RFS F	202
79	Programmatics Matrix	207

80	Technology Development Matrix	208
81	Structural Subsystem Matrix	209
82	TPS Matrix	210
83	Propulsion Matrix	210
84	Probability Relationships Between Budget Alternatives and Technology Development Projects	212
85	Probability Relationships Between Budget Alternatives and Vehicle Subsystems	212
86	Additive Effect, Multiplicative Relationships Between Aerothermal Requirements and Vehicle TPS	213
87	Relationships Between G-limit Requirements and Vehicle Substructures . .	214
88	Additive Effect, Multiplicative Relationships Between Wing Box Technology Development and the Wing Structure	215
89	Relationships Between Composite Tank Technology Development and the Propellant Tanks	215
90	Relationships Between Common Bulkhead Technology Development and Vehicle Structures	216
91	Relationships Between TPS Technology Development and Vehicle TPS . . .	216
92	Programmatics Relationship Matrix	217
93	Tank Structures Probability Relationship Matrix	218
94	TPS Probability Relationship Matrix	219
95	Tank Additive Effect, Multiplicative Relationship Matrix	220
96	Wing Additive Effect, Multiplicative Relationship Matrix	221
97	Monte Carlo Results for Vehicle Weight	222
98	Monte Carlo Results for Vehicle Margin	223
99	Monte Carlo Results for Tank Structures Margin	224
100	Monte Carlo Results for Structures Margin	224
101	Monte Carlo Results for TPS Margin	225
102	Monte Carlo Results for Propulsion Margin	225
103	Monte Carlo Results for Vehicle-Wide Margin	226
104	Scatterplot for Subsystem Margins	227
105	Scatterplot Matrix for Subsystem Margins Highlighting Thresholds	228
106	Monte Carlo Results for Combined Total Weight	231
107	Monte Carlo Results for Combined Total Margin	231

108	Monte Carlo Results for Structural Subsystems	234
109	Monte Carlo Results for Structural Subsystem Margin	234
110	Prediction Profiler for Structural Subsystems–Baseline Requirements	235
111	Scatterplot Matrix for Selected Fuel and Oxidizer Tank Conditions	237
112	Tornado Plot for Structural Subsystem Design Choices	238
113	Scatterplot Matrix for Structures Subsystem Condition Selections Highlighting Cases With Mass Growth of More than 1,437 lbs	240
114	Scatterplot Matrix for Structures Subsystem Condition Selections Highlighting Cases With Mass Growth of More than 1,437 lbs	241
115	Structural Subsystem Margin Having Decided on Composite Tanks and Baseline Wing	242
116	Structural Subsystem Margin Having Decided on Vertical Tail Placement on Body	243
117	Monte Carlo Results for Structural Subsystems, 6 g Limit	244
118	Monte Carlo Results for Structural Subsystem Margin, 6 g Limit	244
119	Scatterplot Matrix for Structures Subsystem Condition Selections Highlighting Acceptable Designs, 6 g Limit	245
120	Monte Carlo Results for Structural Subsystems, Composite Technologies Eliminated	247
121	Monte Carlo Results for Structural Subsystem Margin, Composite Technologies Eliminated	247
122	Comparison Between Baseline Monte Carlo Simulation and Eliminated Composite Technologies	248
123	Prediction Profiler	249
124	Scatterplot Matrix with 6g Limit and Increased Single-Stage Vehicle Performances Cases Highlighted	250
125	Monte Carlo Results for Total Weight Excluding Demanding Requirements	251
126	Monte Carlo Results for Total Margin Excluding Demanding Requirements	251

SUMMARY

One of the biggest challenges in the development of a space or launch vehicle is predicting a vehicle's final performance during the initial phases of design. These predictions are difficult due to the high degree of uncertainty surrounding a conceptual design. Optimistic predictions have led to expensive weight-reduction programs, decreased vehicle capability, and the cancellation of development programs.

A significant driver contributing to the reduction of a space or launch vehicle's performance capability is weight growth. While mass growth will affect nearly all programs, the amount of mass growth is subject to high uncertainty— the historical record shows a large variance in the amount of growth experienced by development programs.

In order to address the uncertainty surrounding mass growth, program managers add an extra mass allotment to the dry mass of a vehicle— this extra allotment is known as a design margin. The design margin is allocated early in the development process. Furthermore, the decision as to how much design margin to allocate is a critical design decision— too much margin will lead to an over-sized system while too little margin leave a development program susceptible to not meeting performance requirements.

A promising method used in different industries to make this decision is known as range estimating. This method involves breaking a project down by its work breakdown structure and assigning a range of uncertainty to each subitem; the sum of each subitem will represent the collective uncertainty for the total system. However, recent research into this method has shown that range estimating is only appropriate for well-defined programs and would not be applicable during early design phases when a vehicle baseline is subject to change.

In order to account for all relevant uncertainties, a hybrid method of forecasting mass is developed. This hybrid methodology merges two separate styles of analyzing a development program's uncertainties. Because range estimating has been established as a mature methodology in multiple fields, it will be used in the hybrid methodology as the bottom-up

analysis. Because range estimating is being used unmodified, the new focus of research focuses on creating a top-down methodology which accounts for baseline changes without double-counting estimates from range estimating.

In order to account for potential baseline changes in a vehicle two problems must be solved: alternative configurations must be identified, and the performance differences relative to the original baseline must be identified. The problem of identifying potential alternative configurations can be solved through the use of morphological analysis. Morphological analysis has been proven to be a method to generate a large number of possible alternatives. However, extracting quantitative data from this large number of possible alternatives is a challenging problem.

In order to extract quantitative information from a morphological analysis, an extension, known as executable morphological analysis (EMA), is developed. The central idea behind EMA is that by including small pieces of information about each condition in the morphological field, then quantitative information can be easily extracted from each combination.

EMA is composed of two data structures: the executable matrix of alternatives and the relationship matrix. The executable matrix of alternatives is an extension of the morphological field. In this extension, each condition contains attributes; specifically it must contain attributes which describe the condition's effect on the model's output and the condition's likelihood of being selected. Similarly, the relationship matrix is an extension of the cross-consistency matrix. The relationship matrix contains relationships between the conditions of the model and implements incompatibilities.

For the specific implementation of EMA for addressing the problem of forecasting margins, each condition will have three pieces of information: a likelihood, a multiplicative effect, and an additive effect. The likelihood value is a specified chance of a condition being selected relative to the other conditions in the same parameter; these likelihood values are constrained so that the sum of all likelihood values in a single parameter is equal to unity. The multiplicative effect acts as a scalar multiplier to the total dry weight of a vehicle, and the additive effect acts as a simple addition to the dry weight.

EMA is implemented through an object-oriented design. This object-oriented framework acts as an enabler of EMA and represents an advancement over matrix-based implementations of traditional morphological analysis because EMA requires state-dependent functionality and new attributes and methods in each level of the data structures. Furthermore, the object-oriented framework can easily enable future applications of EMA through subclassing the abstract classes.

In order to evaluate the computational requirements of evaluating an EMA model, a series of Monte Carlo simulations varying the size of the executable matrix of alternatives and the interconnectedness of the relationship matrix was conducted. This experiment found that the number of Monte Carlo simulation cases necessary to generate accurate probabilistic results is relatively constant with the size of the executable matrix of alternatives and the complexity of the relationship matrix. For the largest fields tested, the expected value and 80% quantile measurements converged in 1250-1500 Monte Carlo cases. This result shows that useful quantitative information can be extracted from an executable morphological analysis without significant computational effort.

With EMA algorithms and data structures developed, sample problems were used to demonstrate EMA's effectiveness as a forecasting methodology. Two sample problems are conducted. The first focuses on the use of EMA to predict changes in a the Space Shuttle Orbiter's dry weight while utilizing historical information and forecasts. The second sample problem focuses on demonstrating the hybrid methodology on the FAST RFS F, an advanced launch vehicle technology demonstrator.

The Space Shuttle Orbiter problem tests the use of EMA as a method of forecasting baseline changes. By using a predetermined percentage to model in-scope mass growth, this sample problem isolates the predictive effect of EMA on forecasting mass growth associated with baseline changes. In order to construct the EMA model for this problem, the historical record of proposed Orbiter designs was used to create a morphological field. The mass-impacts of each potential baseline change was determined by utilizing original mass properties reporting documents from the Orbiter program; the use of estimates based on original documents mitigates hindsight bias and ensures that this experiment produces a

new forecast based on the input of 1970's experts. The output of the EMA model produced very good predictions of the Orbiter's final flight weight.

The second sample problem uses the complete hybrid methodology is used to estimate the design margin for a novel concept– the FAST RFS F technology demonstration vehicle. This sample problem utilized mass properties data from FAST program contractors to inform a weight breakdown structure. This weight breakdown structure formed the basis of a range estimating analysis. This range estimating analysis was augmented with an extensive EMA model; the EMA model for this sample problem was broken into three separate sections: programmatics, technology development programs, and vehicle alternatives. The Monte Carlo simulation output of this EMA model was used to conduct a probabilistic analysis on the total vehicle and subsystem weights as well as a sensitivity analysis and a what-if analysis. Finally, the results of both range estimating and EMA were used to examine the weight growth of the FAST RFS F.

The FAST RFS F sample problem demonstrates the use of morphological analysis to generate alternative vehicle baseline scenarios and demonstrates the use of a hybrid method with both a top-down and a bottom-up analysis component for analyzing vehicle development programs. Finally, this sample problem demonstrates the use of EMA and range estimating as a way to quantitatively analyze uncertainties surrounding a novel project; this demonstration shows that the original research motivation has been addressed and that a hybrid analysis method consisting of range estimating and EMA is a useful tool for analyzing uncertainties and assigning a design margin.

CHAPTER I

INTRODUCTION

In today's world, space-based assets are critical infrastructure— satellites provide communications, remote sensing, radio-based navigation through the global positioning system, and world-wide, coordinated timing. However, before a satellite can perform its mission, it must be carried into orbit on a launch vehicle.

1.1 Launch Vehicles

Modern launch vehicles trace their existence back to the initial research done by Wernher von Braun for the German V-2 program during World War II. After the war, von Braun and 117 other German scientists went to the United States and formed the basis of American research into rocketry. This group conducted research programs on American soil with captured V-2s and their engines. This program led to the development of engines for the XLR83 engine which served as the basis of many future rocket engines which powered the Saturn V, Delta, and Atlas launch vehicles; the propulsion program also led to the founding of the Rocketdyne division of North American Aviation, one of the primary rocket engine manufacturers in the United States. This technological foundation served as the basis for U.S. launch vehicles until the Space Shuttle and the EELV programs. [149]

1.1.1 Current Launch Vehicles

1.1.1.1 Active Vehicles

The current mainstays of the U.S. launch vehicle fleet are the Atlas V and the Delta IV, the two elements of the Evolved Expendable Launch Vehicle (EELV) program. The EELV program emerged in 1995 as a response to the state of U.S. launchers in the 1980s and early 1990s where national security payloads originally intended to be launched by the space shuttle were relegated to legacy expendable launch vehicles. [62, 61] The EELV program was intended to replaced these legacy launchers with two new families of launch vehicles.

Furthermore, this new program was supposed to reduce costs and improve operational flexibility. Finally, the development of two different launch vehicle families was meant to lead to assured access to space as two independent vehicles acts provides a mutual backup capability. [105]

As part of the overall cost savings, the original EELV program was jointly funded between the government and the two winning prime contractors, Lockheed Martin and The Boeing Company. The government provided a total of \$1 billion (\$500 billion to each contractor) in development money while Lockheed Martin spent \$1.6 billion and Boeing spent \$2.3 billion. The prime contractors would recoup their investment by launching commercial satellites. The launching of commercial satellites would also help pay for overhead and launch site operations costs. However, the commercial satellite market did not materialize, and the government was stuck paying wholly for the operations of two families of launch vehicles while the prime contractors wrote down large portions of their investment as losses. [105]

Today, the EELV program is ongoing with both the Atlas V and the Delta IV being manufactured and operated by the United Launch Alliance (ULA), a joint venture between Lockheed Martin and Boeing formed in an effort to reduce costs. However, EELV launch costs are ever increasing, and ULA is currently negotiating a block buy agreement with the government in an attempt to reign in costs. [87] While launch vehicle pricing is extremely difficult to estimate, [105] recent estimates place an Atlas V launch cost in the range of \$150–\$180 million per launch (not including additional subsidies) [87] or approximately \$3,500–\$5,000 per pound to low earth orbit. [71] Similarly, a Delta IV medium has a cost of approximately \$4,000–\$6,800 per pound to low earth orbit. [71]

While the EELV cost savings have not met initial projections, [71] the vehicles have produced an improvement in reliability. Historically, the reliability of launch vehicles is . [118] In comparison, the Atlas V has successfully launched 27 out of 28 missions (the single non-successful mission ended with the two payload satellites reaching target orbits with on-board propulsion) [38], similarly the Delta IV has 27 out of 28 successful missions– its only failure was a demonstration mission of the Delta IV heavy configuration.[39]

1.1.1.2 *Vehicles Under Development*

Because current vehicles are expensive and require extensive processing before launch, the U.S. Air Force has identified a reusable booster system (RBS) as a replacement system for the current EELV fleet. This new system currently has a planned initial operating capability in the year 2025. [51]. This new system will have the economic requirements such as a 50% cost reduction compared to current expendable vehicles as well as operability requirements such as 24-48 hour turnaround, 2-8 hour call-up, and flexible basing.[55] These requirements represent a dramatic shift from what is traditionally thought of as launch vehicle operations which are characterized by lumbering operations and expendability. Furthermore, these operational advancements will enable national security missions such as satellite constellation reconstitution.

The current approach is a hybrid launch system—a reusable first stage and an expendable upper stage.[51] An illustration of this system can be seen in Figure 1. During a nominal launch, the combined system launches vertically, the first stage burns alone or in parallel with an upper stage, the two stages separate, and the first stage carries out a return to launch site (RTL) maneuver. The current baseline RTL maneuver is a rocketback maneuver where the reusable first stages rocket engines are used impart enough energy to the system to enable a glide back to the launch site.[20]

1.1.2 **The Physics of Space Launch**

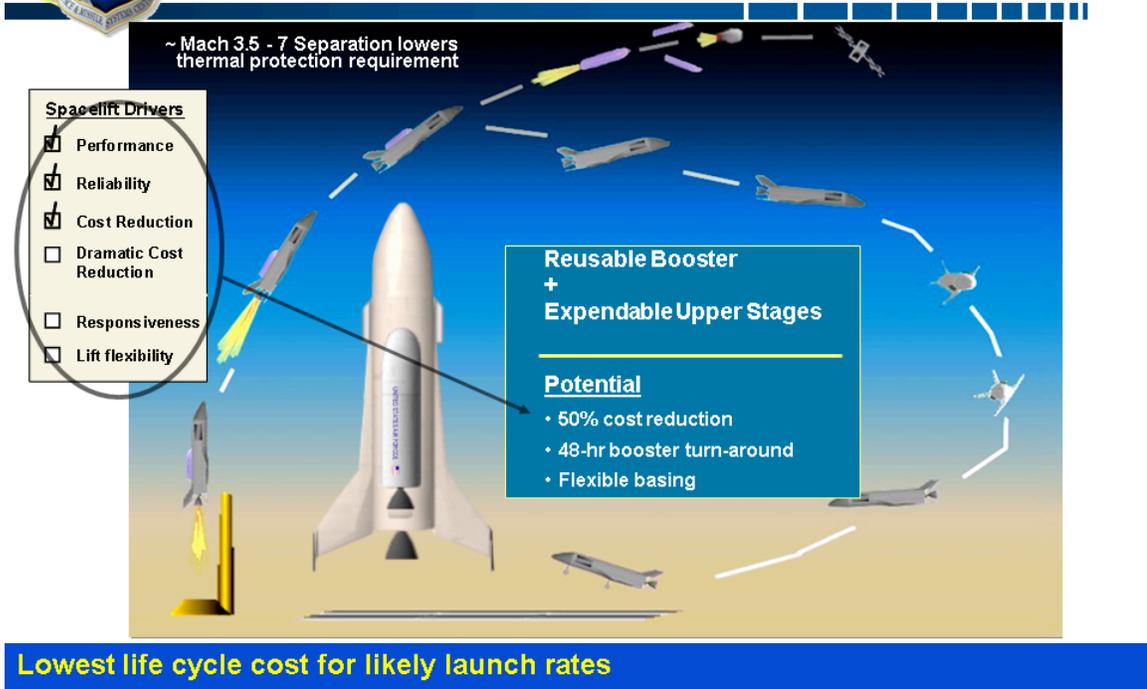
Launch vehicles are expensive and risky to operate because space launch is a very difficult process due to the physical demands placed upon a launch vehicle and payload. The mission of a launch vehicle is to place a satellite in a desired orbit. Fundamentally, this is a mission to add a substantial amount of energy to the spacecraft; the chemical energy stored in the launch vehicle’s fuel tanks is used to accelerate the system to the desired orbit. [71] This chemical energy is transferred into kinetic energy through the use of rocket engines while potential energy is gained throughout the ascent trajectory.[10, 29]

A standard way of expressing the energy of an orbit is through the specific orbital energy—the sum of the potential and kinetic energy per unit mass of the vehicle. The expression



Approved for Public Release

Reusable Booster System (RBS) Concept



8

Figure 1: U.S. Air Force Reusable Booster Concept [51]

for specific orbital energy, ϵ , can be seen in equation 1. [10] Within this equation, the first term, $v^2/2$, represents the specific kinetic energy, and the second term, $-\mu/r$, represents the specific potential energy where μ is the gravitational constant of the planet in question. For Earth, the gravitational constant is $1.40765 \times 10^{16} ft^3/s^2$. [49] Within this potential energy formulation, the datum for zero potential energy is considered to be where an object is no longer in Earth's sphere of influence. For non-thrusting orbits, equation 1 can be reduced to equation 2; within this equation a represents the length of the semimajor axis of the orbital ellipse; for circular orbits, the semimajor axis length is equal to the radius of the circular orbit. [29]

$$\epsilon = \frac{v^2}{2} - \frac{\mu}{r} \quad (1)$$

$$\epsilon = -\frac{\mu}{2a} \quad (2)$$

While specific orbital energy is a good measure for conventional satellites in orbit around a single body, a similar measurement, C3 or characteristic energy, is used to describe the energy required to send a spacecraft on an interplanetary trajectory. [29] For an interplanetary mission, a spacecraft launched from earth must enter a hyperbolic orbit which, upon leaving Earth's sphere of influence, enters the correct sun-centered interplanetary transfer orbit. The velocity of the spacecraft when it leaves Earth's sphere of influence is known as the hyperbolic excess speed or v_∞ . [10] Unlike traditional satellites where the specific orbit required can be analytically determined years ahead of time, the v_∞ is a function of the geometry of the solar system at the time of launch. Because of this issue, interplanetary mission designers use C3 to measure launch vehicle capability. C3, as expressed in equation 3, measures the maximum energy that a launch vehicle can impart to a payload of a given mass and is equal to twice the specific orbital energy. [29]

$$C3 = v_\infty^2 \quad (3)$$

Many orbits of interest and their corresponding physical characteristics can be seen in

Table 1: Specific Orbital Energies of Common Orbits

Orbit	Altitude (at apogee)	Orbital Speed (at apogee)	Specific Orbital Energy
Low Earth	100 miles [29]	25,615 ft/s	$-3.28 \times 10^8 \text{ ft} \cdot \text{lb}/\text{slug}$
	600 miles [29]	24,171 ft/s	$-2.92 \times 10^8 \text{ ft} \cdot \text{lb}/\text{slug}$
Molniya	25,000 miles [29]	4865 ft/s	$-8.02 \times 10^7 \text{ ft} \cdot \text{lb}/\text{slug}$
Semi-synchronous	12,570 miles [120]	12,698 ft/s	$-8.06 \times 10^7 \text{ ft} \cdot \text{lb}/\text{slug}$
Geostationary	22,241 miles [29]	10,086 ft/s	$-5.08 \times 10^7 \text{ ft} \cdot \text{lb}/\text{slug}$

Table 1. The lowest energy orbit is that of a ballistic missile which have an orbit that intersects the surface of the earth at the intended target. [10] Low earth orbit (LEO) describes the family of orbits between 100 and 600 statute miles [29]– below the Van Allen radiation belts. [143] These orbits are used by the space shuttle and international space station as well as a large number of earth-observing satellites.[10, 143] Spacecraft, such as the GPS satellite constellation, which complete two orbits of the earth per day are in semi-synchronous orbits.[69] Geosynchronous orbits have an orbital period equal to a single sidereal day. A special case of geosynchronous orbits is the geostationary orbit; geostationary orbits have 0° orbital inclination allowing them to remain fixed over a specific point on the equator. This characteristic makes geostationary orbit a very popular orbit for communications and earth observing satellites. However, geostationary orbits do not provide good coverage of high latitude regions of Earth, such as the Russian Federation. [143] In order to solve this problem Russian (formerly Soviet) engineers placed satellites in a Molniya orbit. Molniya orbits are highly elliptical and have a 63.4° inclination; this inclination allows the apse line, the central line of an ellipse, to remain stationary with respect to the ground. [29]

Because missions to place objects in specific orbits involve accelerating a payload to the point where it has sufficient orbital energy, the primary measure of a launch vehicle’s performance is the amount of acceleration it can provide– the total acceleration is also known as the change in velocity or ΔV . [57, 143] Typical missions place a payload into a parking orbit– usually in LEO. From this parking orbit, the upper stage of the launch vehicle conducts a second burn to place the vehicle in a transfer orbit. Finally, once the spacecraft reaches its desired orbit uses it onboard propulsion systems to perform a circularizing burn

and conduct an orbital plane change maneuver (if necessary). Each one of these burns changes the specific energy of the spacecraft and therefore requires a specific ΔV . Depending on the mission, trades can be done between the ΔV required to perform orbital transfers and the time necessary to perform orbital transfers. [143, 29] Furthermore, the desire for launch sites to be located as close to the equator as possible can be seen from this formulation. As the orbital speed is calculated in the Earth-fixed celestial frame, the initial velocity of the launch site due to the Earth's rotation contributes to the velocity required to reach orbit reducing the ΔV requirement of the launch vehicle. [127] The maximum initial velocity which can be attained is 1,522 ft/s at an equatorial launch site. As the United States Eastern Range in Florida is at 28.5° N, vehicles launching from this range have an initial velocity of 1,337 ft/s.

1.1.2.1 *The Rocket Equation*

The transfer of a rocket's chemical potential energy into ΔV can be mathematically modeled by the rocket equation. [49, 57, 143] This equation captures the structural efficiency of the vehicle, the amount of fuel burned, and the efficiency of the propulsion system. The simplest form of this equation, the modeling of a single stage rocket or a single burn of an in-space propulsion system, can be seen in equation 4. [116, 49]

$$\Delta V_{effective} = g_0 I_{sp} \ln \frac{m_{initial}}{m_{final}} \quad (4)$$

Equation 4 is an analytical expression for the total effective ΔV generated by a single burn, but the effective ΔV can be broken down into the ideal ΔV and ΔV losses as can be seen in equation 5. The ideal ΔV represents the ΔV required to change orbits or conduct an in-space maneuver such as an orbital plane change. [143] Losses are introduced into the burn through three mechanisms: aerodynamic drag, gravity, and thrust control. Drag losses occur when the vehicle is flying through the atmosphere and has to use a portion of the rocket's thrust overcoming atmospheric drag. Gravity losses occur because a vehicle launches vertically and needs to use its thrust overcoming Earth's gravity. Both of these losses can be mitigated through selection of the vehicle's thrust-to-weight ratio and

Table 2: Typical Values of I_{sp} [143]

Propellant Combination	I_{sp} (sec)
O ₂ and RP-1	350
O ₂ and H ₂	450
N ₂ O ₄ and N ₂ H ₄	330
Solid Motors	280-300

optimization of ascent trajectory. [57, 143] For most launch vehicles gravity losses tend to be from 2,460 ft/s to 4,900 ft/s while drag losses tend to be between 65 and 130 ft/s. [143] Thrust losses occur when the thrust vector is not aligned with the velocity vector as is the case when the guidance system is actively steering the vehicle. [49]

$$\Delta V_{effective} = \Delta V_{ideal} + \Delta V_{thrust\ losses} + \Delta V_{drag\ losses} + \Delta V_{gravity\ losses} \quad (5)$$

Within equation 4, I_{sp} is the specific impulse of the vehicle. This term represents the efficiency of the propulsion system. I_{sp} is mathematically modeled by equation 6; based on this equation, the efficiency captured by I_{sp} is the thrust of the propulsion system per unit mass flow through the rocket nozzle. For engineering purposes, the thrust per unit mass flow rate is then normalized by the gravitational acceleration of earth. [57, 49] The I_{sp} of typical propellant combinations can be seen in 2. It is worth noting that current chemical propulsion is on the order of 97-98% efficient, and future improvements in efficiency will come at great expense. [71]

$$I_{sp} = \frac{T}{\dot{m}g} \quad (6)$$

The mass ratio, $m_{initial}/m_{final}$, represents the total amount of fuel used during a burn. If all of the propellant is used, as would be the case during a launch vehicle's mission, the mass ratio also represents the mass efficiency of the vehicle. [116] Within the rocket equation, taking the natural log of the mass ratio represents the fact that thrust applied towards the end of a burn when propellant has been burned off will have a much greater impact on ΔV when compared to thrust applied towards the beginning of a burn. Additionally,

the response of the natural log heavily penalizes inefficient mass use— as the mass ratio decreases, the ΔV penalty increases exponentially. [49, 57, 143]

Structural mass efficiency can be represented through the structural coefficient as seen in equation 7; another key measure is the payload ratio— the ratio of the payload weight to the structural and propellant weights, equation 8. Using both the structural coefficient and the payload ratio, the mass ratio can be expressed by equation 9. Based on this equation, a trade between payload weight and structural weight exists because both the payload ratio and the structural efficiency coefficient appear in the denominator of equation 9. [49]

$$\epsilon = \frac{m_{structure}}{m_{propellant} + m_{structure}} \quad (7)$$

$$\lambda = \frac{m_{payload}}{m_{initial} - m_{payload}} \quad (8)$$

$$MR = \frac{1 + \lambda}{\epsilon + \lambda} \quad (9)$$

While the terms in the rocket equation can be analyzed separately, the impact on performance due to propulsion efficiency and mass efficiency are highly coupled. For example, inefficiencies in the propulsion system will require more propellant to provide the same amount of ΔV as an originally closed vehicle. This increase in fuel will in turn require an increase in the mass of the propellant tank structure in order to accommodate the additional propellant. This increase in total mass will require additional propellant in order to lift the additional structure and propellant mass— this feedback loop will continue until the vehicle re-closes. Additionally, this increase in mass may require a change to the propulsion system in order to increase the thrust-to-weight ratio of the vehicle; a change to the propulsion system will likely also increase the dry mass of the vehicle. These positive feedback loops can quickly increase the gross liftoff weight of a vehicle and are the primary reason why single-stage-to-orbit vehicles are infeasible at the existing level of technology. [116, 36]

1.1.2.2 Staging

Because of this sensitivity to mass, it is advantageous to use multi-stage rockets. A single stage of a launch vehicle is comprised of rocket engines, propellant tanks, and the necessary subsystems required to operate the propulsion system (vehicle-wide functions such as avionics are shared by all stages). The total propellant necessary to lift a given payload is effectively distributed throughout the stages of a launch vehicle, and a stage is jettisoned when its propellant has been used. The jettisoning of stages minimizes the total mass required to be accelerated to orbital speed. [49, 143]

In addition to the benefits to the overall mass ratio, multi-stage launch vehicles benefit from the ability to use different propulsion systems on different stages. First stages which are responsible for liftoff and initial acceleration require large amounts of thrust to achieve a sufficient thrust-to-weight ratio. Propulsion systems capable of producing this amount of thrust, typically solid rocket motors and O₂/RP-1 liquid engines, are far less efficient than the O₂/L₂ powered upper stages. [49, 143]

When analyzing a multi-stage vehicle, the rocket equation can be applied to each individual stage. For a given stage, the payload is the remainder of the launch vehicle in addition to the payload to be delivered to orbit; the equation for a stage's payload ratio can be seen in Equation 10. The given mass ratio for a single stage burn can be approximated using this payload ratio and a stage's structural coefficient, Equation 11, as inputs to Equation 12. Finally, this result can be used in Equation 13 to get the total ΔV generated by a rocket stage; the total ΔV generated by the launch vehicle is the sum of the ΔV produced by each stage. [49]

$$\lambda_n = \frac{m_{initial_{n+1}}}{m_{initial_n} - m_{initial_{n+1}}} \quad (10)$$

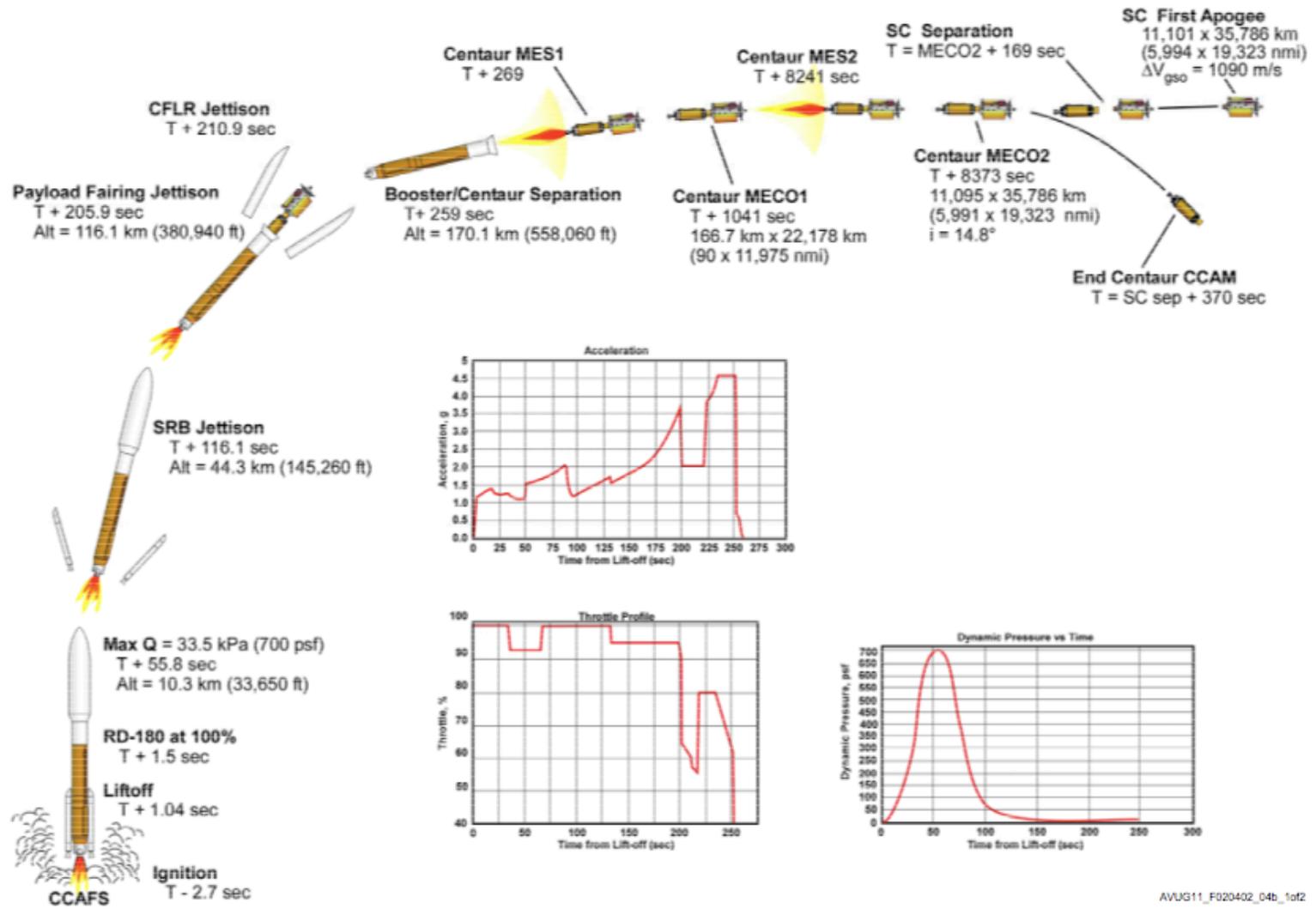
$$\epsilon_n = \frac{m_{structure_n}}{m_{initial_n} - m_{initial_{n+1}}} \quad (11)$$

$$MR_n = \frac{m_{initial_n}}{m_{final_n}} \simeq \frac{1 + \lambda_n}{\epsilon_n + \lambda_n} \quad (12)$$

$$\Delta V_{effective_n} = g_0 I_{sp_n} \ln MR_n \quad (13)$$

While these equations model a traditional multi-stage rocket such as a Saturn I or Titan II, modern launch vehicles take advantage of parallel staging. In parallel staging, two or more stages of the rocket will burn concurrently and will burn out at different times. This approach can provide many benefits such as allowing the upper stage engine to be started on the ground (Space Transportation System) or adding strap-on solid rocket motors to increase the total impulse of the vehicle (Delta II, Delta IV, and Atlas V). An example of a modern parallel staging launch vehicle's flight profile can be seen in Figure 2. [49]

Figure 2.4.2-4: Typical Atlas V 521 Extended Coast GTO Ascent Profile and Ground Trace (1 of 2)



AVUG11_F020402_04b_1of2

Cleared for Public Release Through OSR

Figure 2: Atlas V 521 Launch Profile [134]

1.2 Uncertainty in Space and Launch Vehicles Performance Prediction

As was shown in section 1.1.2.1 the ability of a launch vehicle or satellite to achieve an orbit or perform a mission is due to the mass efficiency and propulsion efficiency of a vehicle. However, during conceptual design when vehicles and space missions are being designed, the uncertainty surrounding performance parameters and weight estimations is high. History has shown that early predictions of mass efficiency or propulsion efficiency tend to be optimistic. [116]

1.2.1 Mass Growth in Space and Launch Vehicles

1.2.1.1 Launch Vehicle Mass Growth

A report from NASA's Marshall Space Flight Center (MSFC) details the mass growth of MSFC-led launch vehicles. [23] This data can be seen in Table 3. This data covers the major development programs in human spaceflight history– the Saturn booster series and the Space Transportation System. Within this list, the Saturn series and the ET are liquid systems while the SRB and the IUS are solid rockets; additionally, all of these systems except for the STS SRB are designed to be expendable. For the Saturn V stages, #501 is the first Saturn V launched while #506 is the rocket used for the Apollo 11 mission after having undergone a weight reduction program.

Within this list, the largest sources of growth occur in subsystem layouts for the STS. The IUS's airborne support equipment more than doubled in size while the SRB subsystems increased by 45%. Individual stages had less overall growth, but still experienced a large variation in percentage growth– between 3% and 24%. Furthermore, the large growth in the STS systems shows that predicting mass growth has not greatly improved between the Saturn and STS programs.

1.2.1.2 Space Vehicle Mass Growth

Heineman out of NASA's Johnson Space Center (JSC) reported the mass growth of JSC-lead manned spacecraft. The percentage mass growth with respect to the initial estimate at conceptual design. [54] As with the list of launch vehicles, this list contains both the first

Table 3: Launch Vehicle Historical Mass Growth [23]

Launch Vehicle	Percentage Mass Growth
Saturn I, S-1 Stage	16 %
Saturn I, Interstage	24 %
Saturn I, S-IV Stage	16 %
Saturn V, S-1C Stage	#501-7 % #506-3 %
Saturn V, S-II Stage	#501-32 % #506-19 %
Saturn V, S-IVB Stage	#501-33 % #506-24 %
Space Transportation System, ET (SWT)	13 %
Space Transportation System, SRB (without DFI)	13 %
Space Transportation System, SRB Sub-systems	43 %
Space Transportation System, Interim Upper Stage (IUS)	11 %
Space Transportation System, Interim Upper Stage Airborne Support Equipment	122 %

Table 4: Manned Spacecraft Historical Mass Growth [54]

Vehicle	Percentage Mass Growth
Mercury Capsule	28 %
Gemini Capsule	18 %
Apollo Command Module	22 %
Apollo Lunar Module	50 %
Skylab	56 %
Space Transportation System Orbiter	25 %

attempts at manned spacecraft as well as Orbiter. Similar trends can be seen from this list—a wide range of possible mass growth (ranging from 18% to 56%) as well as no significant improvement between early programs and the Orbiter.

Hawkins reported the mass property information of fourteen U.S. Air Force satellites, one U.S. Navy satellite, two commercial satellites, and a NATO satellite. The dry mass as a function of program completion can be seen in Figure 3. [53] Like the launch vehicles and manned spacecraft, this chart shows a large spread of potential mass growth (13% to 55%). A similar study was conducted by Bitten et al. which reported on the mass growth experienced by NASA science missions. The growth curves for each individual mission as well as the average mass growth can be seen in Figure 4. [16] The space missions in this figure launched between the years 2000 and 2009 making this data significantly more recent than Hawkins’s study. However, the trends in mass growth are nearly identical even though

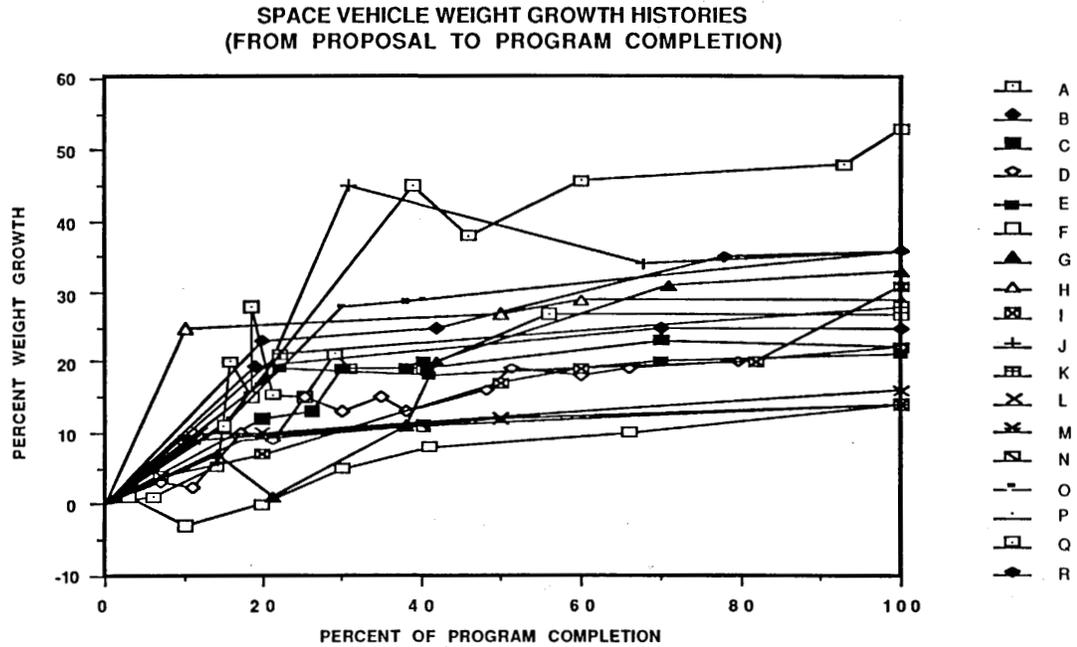


Figure 3: Dry Mass Growth of Satellites [53]

these space vehicles are generations ahead of the vehicles reported in Hawkins's study. This supports the conclusion that even though technology has progressed, the ability to successfully predict mass growth has not improved with time.

1.2.1.3 Advanced Concept Mass Growth

Another area where mass growth has been very visible is in advanced programs. Table 5 contains a list of four historical space plane concepts which experienced mass growth. [130]

The first concept on this list is the X-20 Dyna-Soar, an Air Force space plane designed to conduct lifting re-entry research. The original design called for it to be launched on a man-rated Titan II booster. However, mass growth due led to the need to switch to the newly designed Titan III rocket; this launch vehicle switch would require an expensive series of tests to man-rate the Titan III. This new expense partially led to the cancellation of the X-20 program in 1963. [145]

Perhaps the greatest example of mass growth is the X-30, National Aerospace Plane (NASP)– a single stage to orbit (SSTO) development program. Originally estimated at a gross mass 50,000 lbs, development cycles led to a gross mass of 450,000 lbs. This massive

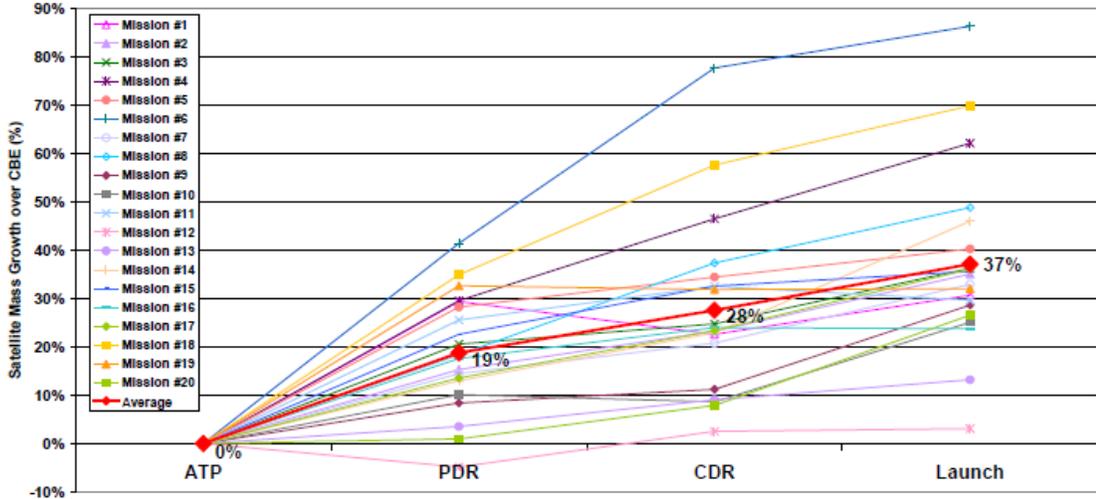


Figure 4: Dry Mass Growth of Space Missions [16]

mass growth in conjunction with technological hurdles led to the cancellation of this program in 1990. [130]

The next program working towards an SSTO was the X-33 program. This was a demonstrator vehicle designed to test technologies such as composite propellant tanks, metallic thermal protection systems, and aerospike engines. Ultimately, the mass-reducing technologies did not perform as expected— for instance, composite propellant tanks failed and needed to be replaced with heavier tanks. The X-33 program eventually ended with 75% of the vehicle assembled in March of 2001.[76, 130]

The only member of this list to actually become operational is the X-37. This reusable spaceplane has made two flights to orbit after being lofted by an Atlas V 501 launch vehicle. This vehicle, like the X-20, serves as a testbed for technologies related to reusable spacecraft as well as operations necessary to the turnaround of reusable vehicles; additionally the spaceplane serves as a satellite bus to test new sensors and equipment. The first flight of the X-37b orbital test vehicle in 2010 demonstrated an autonomous landing after a 224 day orbital mission. [145, 48]

1.2.1.4 Rocket Engine Mass Growth

While much can be gained by looking at completed vehicles, additional insight can be gained by looking at particular subsystems. The most complicated subsystem on a launch vehicle

Table 5: Advanced Concept Mass Growth [130]

Vehicle	Percentage Mass Growth
X-20	33 %
X-30	800 %
X-33	45 %
X-37	25 %

Table 6: Rocket Engine Historical Mass Growth [144]

Engine	Percentage Mass Growth
F-1	59 %
J-2	69 %
SSME	14 %
RS-68	13 %

is the rocket engine. During large scale development programs, the engine development is usually run as a separate development program to the stage on which it will be mounted. Furthermore, engine development can be started before its stage is conceived, and engines can be used on multiple stages.[148, 14]

White reported on the historical growth of Rocketdyne engines. [144] Within this sample of engines, the J-2 and F-1 engines were initially developed independently of their stages and suffered mass penalties due to requirements changes and vehicle integration issues. Furthermore, each engine underwent weight reduction programs resulting in a 13% to 16% reduction in weight. The final percentage increases in mass over proposal mass properties can be seen in Table 6. [144, 148, 14]

1.2.1.5 Mechanisms of Mass Growth

The previous sections have demonstrated that mass growth is a ubiquitous problem in the development of space vehicles. It is also useful to examine exactly why this growth occurs in the context of a vehicle development program.

The AIAA Mass Properties Control Standard has nine standardized categories which can be used to describe a change in the mass properties of a vehicle.[6] These nine categories and their descriptions can be seen in Table 7. In addition to the AIAA standard, other authors have made similar categorizations of mass growth; both Hawkins [53] and Mathews

Table 7: AIAA Mass Change Codes and Definitions [6]

Category	Description
1) Better Definition of Design	Changes that occur as the design progresses beyond the proposal stage and the design criteria and requirements become better defined. These changes are generally early in the program (prior to drawing release).
2) Out of Scope Changes	Scope changes caused by the procuring authority adding new or changing existing requirements.
3) Redesign	Changes to the original component or subsystem design due to repackaging, failure of a component during testing, impact of other subsystem changes, etc.
4) Maturing Component Design	Updated mass analysis due to the drawing revisions at or after original release. (Item #1 generally relates to mass analysis prior to drawing release.)
5) Error in Previous Estimate	Changes due to errors in the mass calculations for an original or later estimate.
6) Uncontrolled Vendor Changes	If none of the other change codes apply, then this category is a catchall for supplier mass changes over which the contractor has very little control.
7) Mass Reduction Activity	Changes due to official mass reduction efforts.
8) Measured vs. Calculated	Changes due to actual measured mass of components being different from the latest calculated values.
9) Cost Reduction or Schedule Reduction, Added Mass	Mass increases that were incurred to save time and/or money, e.g., substitution for expensive exotic materials, redesign of elaborate machined parts and cutouts to reduce machinery costs, etc.

[82] have ten and eight point categorizations of mass growth respectively. Additionally, these large lists can be broken down into two generalized sources of growth: out-of-scope and in-scope growth. [130, 82]

The first overarching category identified as a mechanism of mass growth is out-of-scope changes; these are simply defined as a change in requirements by the procuring authority.[6] These changes can demand a vehicle with more capability. Additional capability will require a mass allocation for the instrument or payload, and the mass of the vehicle will also increase as this capability will require additional power, fuel, and structure to accommodate this new requirement.

A more complicated category of growth mechanisms is in-scope growth or design uncertainties. The class of growth describes seven of the nine AIAA-defined categories of mass growth (Table 7 numbers 1,3,4,5,7,8, and 9).[6] Another sub-categorization of this category was created by Williams who defined four sub categories: growth due to design specifics (i.e. changes to geometry), growth due to immature technology, and growth due to unknown problems (unknown unknowns). [147] Williams’s categorization is more general and focused on explaining the nature of mass growth; therefore this categorization can be used as a starting point to further explore the nature of mass growth due to design changes.

The first category of in-scope growth is due to changes in the design. Specifically, Williams describes this category by posing the question, “How firm are the geometry, area, thicknesses, etc.?” [147] Put another way, this category of growth occurs because not all parts of the vehicle reach the same level of maturity at the same time. Subsystems which may only be sketched out during the initial mass estimations change as designers fully define these systems. Similarly, subsystems which have underwent preliminary design during early mass estimations are subject to redesign to accommodate newly-defined subsystems.

Another aspect of design maturation concerns the knowledge of the design and the induced environment. Initial mass estimations are made when the knowledge of the design is at a minimum, and the vehicle’s design will have to change as more detailed analyses are conducted. These more detailed analyses will define loading conditions, heating rates, total heat soak, aerodynamic performance, etc. These analyses can result in additional mass due to the need for beefier structures, additional TPS, or redesigned aerodynamic surfaces.

A third category of in-scope growth pertains to the planned inclusions of new technologies in a design. New technologies, defined as technologies which have not yet been deployed in an operational vehicle, are included in a design to add capability or save on overall cost, mass, or power. Because of the inclusion of new technologies, a portion of the development effort will be spent on increasing the technology readiness level (TRL) and manufacturing readiness level (MRL). [81, 33] As the new technologies increase in TRL and MRL, the originally estimated benefits tend to fall short of the original, optimistic estimates. [130] A realization of a technology which does not deliver the intended performance benefits will

result in a net increase in mass.

Another aspect of including new technologies in a vehicle is the development cost to bring up to a TRL of 9 and an MRL of 10. Studies have shown that vehicles which include new technologies tend to encounter development difficulties which inflate the cost of a development program. In order to control costs, some technologies may be omitted from a vehicle design during the development process. [136, 137, 138] The omission of technologies will lead to a mass increase as more mature options will not be as mass efficient; this act of adding mass to account for cost and schedule increases is specifically called out in the AIAA Mass Properties Control Standard (#9 in Table 7). [6]

A final category of growth related to the design of a vehicle involves growth due to unknown problems. [147] While unknown unknowns are present in all development projects, they are exacerbated in novel development programs. As problems arise, it is likely that a mass allocation will be necessary to mitigate the issue.

1.2.2 Propulsion

While not as well documented as mass growth, uncertainty in propulsion has both first and second order effects on the predicted performance of a launch vehicle. As seen in equation 4, the I_{sp} of a propulsion system has a large influence in the effective ΔV produced by a rocket. A deficiency in the predicted I_{sp} will significantly impact the ability to deliver mass to orbit or, in the case of satellites, decrease the lifespan of a satellite.

A prominent historical example of a development program overpredicting I_{sp} is the space transportation system (STS) program or space shuttle. The space shuttle has two main propulsion systems— the space shuttle main engines (SSME) and the solid rocket boosters (SRB). The vehicle configuration is that of a two stage to orbit (TSTO). The first stage involves a parallel burn of both the SSMEs and two SRBs; after two minutes, the SRBs are jettisoned and the orbiter and external tank continue to ascend under SSME power. The next staging event occurs when the external tank is jettisoned and the SSMEs are shut down. The second stage of the TSTO configuration involves the orbital maneuvering system of the orbiter performing a circularizing burn.[61] Of these main propulsion systems, the

SSME's I_{sp} was 2.5 s and the SRB's I_{sp} was 1.5 s short of of the original estimation [115] [116]

Second order effects involve the thrust output of rocket engines and motors. A deficiency in predicted thrust will result in lower thrust-to-weight ratio of the vehicle at launch. This lower ratio will increase ΔV losses due to gravity because the vehicle will spend more time during ascent fighting the effect of gravity during vertical launch and ascent.

1.2.3 The Use of Margin

As has been shown, vehicles will grow in mass throughout the development process. This mass growth leads to non-trivial decreases in the performance of space systems due to the physics of spaceflight. While it is known that a vehicle will increase in mass or otherwise not meet conceptual level performance predictions, the degree of this under-performance is very uncertain. In order to deal with this uncertainty, program managers add an extra mass allotment to the overall dry mass of a space vehicle— this extra allotment is known as the design margin. [5, 6, 32, 23, 125, 150, 131, 130, 89]

A design margin is defined as: “an unallocated portion of a quantifiable, allocable requirement value that is used by design engineers and management to avoid and resolve problems that arise in a system development.” [47] A similar term, often used interchangeably, is contingency. Contingency is defined:

Weight factor included to account for lack of detail or unknowns in the design. It is a function of design maturity and will reduce to near zero when all design details become either calculated or actual weights. Can also be considered as a measure of the error of the weight estimate. [121]

The AIAA Recommended Practice for Mass Properties Control for Satellites, Missiles, and Launch Vehicles defines a mass growth allowance as:

The Mass Growth Allowance is the predicted change to the Basic Mass Properties of an item based on an assessment of the design and the fabrication status of the item, along with an estimate of the design changes that may occur. The design changes may be implemented in order to satisfy the contracted design requirements during the development process. The Mass Growth Allowance associated with these design changes compensates for the lack of design maturity. Configuration changes driven by major contract or requirements changes are not included in the mass growth allowance. [5]

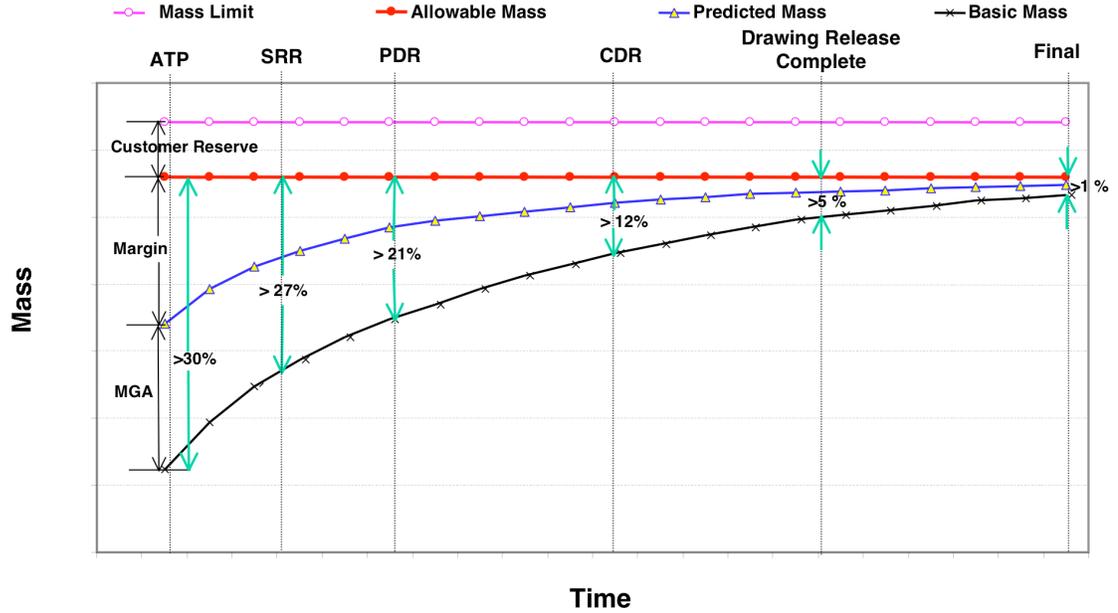


Figure 5: Margin and Weight Growth vs. Time During a Development Program [6]

Within the AIAA’s nomenclature, this definition is complimentary with the term “design mass margin.” The AIAA Standard for Mass Properties Control for Space Systems defines design mass margin as: [6]

The mass margin is meant to mitigate potential mass increases from omissions or refinement of existing design requirements. Mass margin must cover the upper limit of the sum of the predicted mass and the mass uncertainty. [6]

An example of a development program which follows the AIAA nomenclature can be seen in the Mass vs. Time plot in Figure 5. The top horizontal line of this chart is the mission limit weight (also known as not-to-exceed weight); this weight limit is usually defined by the customer but can also represent derived requirements due to technical performance measures. The next horizontal line is the allowable mass– this is the limit to which all margins are calculated. This limit is defined early on in the requirements process and is intended to remain constant throughout the development program. The design margin and mass growth allowance are also called out on this plot; the relationship between the basic mass, allowable mass, design margin, and mass growth allowance is expressed in Equation 14. [6, 121]

$$MGA\% + Margin\% = \frac{MGA + Margin}{Basic\ Mass} \times 100\% = \frac{Allowable\ Mass - Basic\ Mass}{Basic\ Mass} \times 100\% \quad (14)$$

The terms design margin, contingency, and mass growth allowance all refer to the same general concept of adding an additional mass allocation to a vehicle’s empty weight to account for uncertainty. Additionally, terms such as “management reserve” represent additional mass allocations which are also added to the empty weight. [121] For the remainder of this document the term “design margin” will be used to refer to the total margin between the current best estimate (CBE) of mass and the allowable mass. The design margin is defined by Equation 15. [131] This design margin accounts for both in-scope growth due to design maturity and out-of-scope growth due to changes in the design or requirements.

$$\%margin = \frac{allowable\ mass - CBE}{CBE} \times 100\% \quad (15)$$

While the design margin is a system-level metric, individual parts and subsystems have portions of the design margin allocated to them.[131] This withholding of margin on a subsystem level allows the design to progress as individual teams can work with specific mass allocations when sizing structures, power systems, or hydraulic systems.

A key aspect of design margin is that it should be depleted by the end of the program. As seen in Figure 5, the design margin is depleted to within a few percentage points as the vehicle enters production. However, margin depletion to within 1% is an idealized case. Thompson has shown that 30% of historical programs exceed the AIAA recommended design margin. [130] Exceeding the design margin may require depleting the customer reserve— this leads to not being able to fulfill the original requirements of the program. Another alternative to depleting customer reserve is to enact a system-wide mass control (and possible mass reduction) program; these programs are often expensive and significantly delay the development of a new vehicle.[124]

While a mass overrun can lead to an expensive weight control program or possibly even act as a showstopper, the solution to mass overruns is not to allocate a liberal design margin.

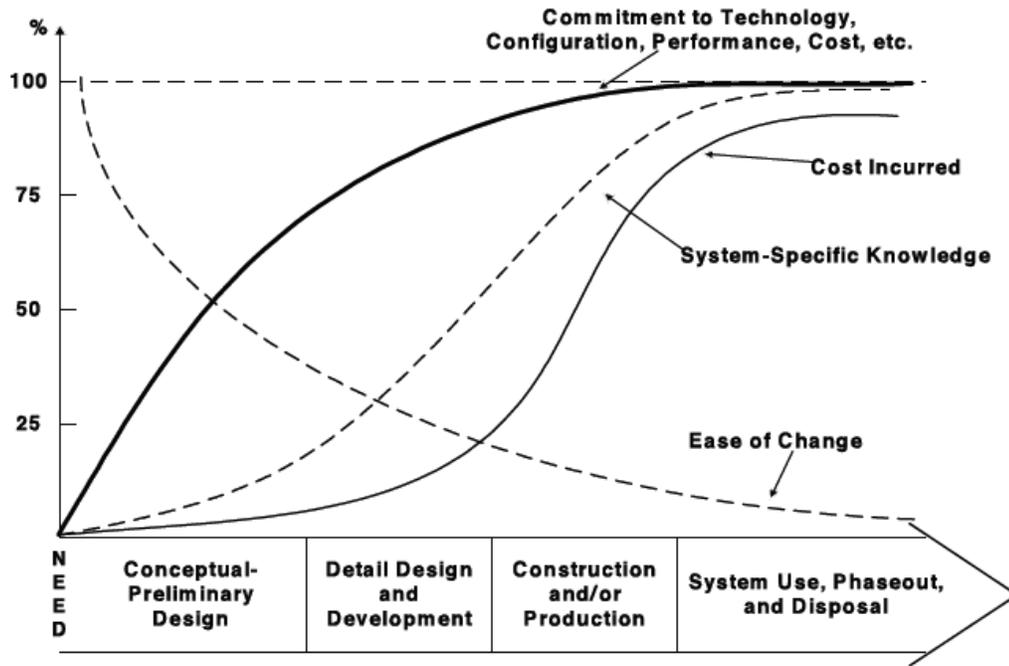


Figure 6: System Commitment, Ease of Change, System-Specific Knowledge, and Cost Incurred vs. Time During a Development Program [17]

As outlined in Section 1.1.2.1, the additional empty weight percentage accounted for in the margin will lead to additional propellant and structural weight. This can quickly lead to a very large system. [77, 36] Systems with a large design margin can be oversized compared to the customer’s original mission requirements. These oversized systems will likely be more expensive to a customer over a vehicle’s lifecycle than a weight control/reduction program. [124]

Because a development program has consequences for both underestimating and overestimating design margin, the decision as to how much design margin to carry during development is one of the most important decisions that program management can make. At the same time, this decision is made early on in the requirements development phase when, as can be seen in Figure 6, knowledge of the vehicle’s future configuration and performance is at a minimum.[17]

This decision can be categorized as risky. Risk in this case can be defined as “the implications of the existence of significant uncertainty about the level of project performance

achievable.” [25] In this definition, the term ‘implications’ refers to the potential consequences of the risk– in this case an over or under-estimation of design margin. The second part of this definition refers to the uncertainty in the total level of performance attainable.

1.2.4 Observations

This discussion of the nature of mass growth and design margin motivates an observation about the nature of mass growth:

Observation 1: The decision to determine how much design margin is carried is a critical design decision which can significantly contribute to the success or failure of a mass-sensitive vehicle.

The first major part of this observation is that the amount of design margin carried can significantly contribute to the success or failure of a vehicle. Successful vehicles such as the Saturn V family were able to achieve the necessary performance through a combination of ample margin and weight-reduction programs. The performance of this system eventually allowed much more capable missions to be flown towards the end of the Apollo program; these missions featured longer surface durations and a lunar rover.

A lack of design margin has contributed to the cancellation of programs. Most recently this can be seen with the Constellation program– the Ares I/Orion stack did not have the design margin necessary to facilitate the original development plans. This led to additional cycles of redesign adding additional cost and time delays to the program. [11] Similarly, the Ares V underwent a significant redesign during its conceptual phase to account for mass growth. This redesign added an additional half-segment to the solid rocket boosters and added a 6th RS-68 engine. [89] Eventually, the Constellation program and the corresponding Ares rocket family was canceled due to the growing costs of the architecture.

The second part of this observation is that the amount of margin to carry is a critical design decision. The criticality of this decision is partially a result of the fact that the amount of margin carried can lead to the success or failure of programs. Furthermore, the design margin assumptions will directly contribute to the sizing of the vehicle because the design margin represents additional mass which must be taken into account. Finally, the decision as to how much design margin is carried is made during the earliest phases

of a development program. Figure 5 shows that a margin should be in place at authority to proceed (ATP); an allocation of margin at ATP is also called out in several standards. [6, 32, 150] Decisions made at this early stage of design will be subject to high amounts of uncertainty.

While this is a critical design decision in traditional satellite or launch vehicle development programs, the decision becomes even more important when working with novel concepts. With novel concepts, the uncertainty surrounding the decision is higher as there is no experience base. Furthermore, novel concepts in space transportation vehicles tend to focus on re-usability (Shuttle, X-20, X-30, X-33, X-37), and these vehicles are more sensitive to additional mass than expendable launch vehicles or single-use satellites. As was shown in Section 1.2.1.3, many novel development programs were canceled in part due to mass growth. The failures of these novel development programs leaves the state-of-the-art of launch systems to expendable launch vehicles that can trace their lineage to the original development programs of the 1950s and 1960s.

The Space Transportation System is an example of a novel concept development program which produced a flight vehicle. However, like many other novel development concepts, the lack of margin on the Shuttle contributed to problems with the vehicle. At authority to proceed, the Orbiter had just over 10,000 lbs of mass margin; this margin was depleted in 1975 while the system mass continued to grow. [54] This led to a degradation of performance—the original payload goal of 60,000 lbs to a 100 nm circular orbit at 28.5° inclination was reduced to 55,250 lbm. [61] This eventually led to the complete redesign of the external tank in order to deliver ISS modules to a 51.6° orbital inclination. [103]

This history of novel concept development motivates observation 2-A:

2-A. A lack of design margin in previous novel concept development programs has led directly to increased costs and can lead to the cancellation of a program.

The contrapositive of this observation provides further insight into the nature of mass growth and mass margin: in order to avoid cancellation or increased costs, a novel concept development program should have ample design margin. In this phrasing, the word “ample” is used to denote the idea that the amount of design margin should be in proportion to the

mass growth that occurs— too much design margin can lead to an unaffordable system. However, as a development program cannot know the eventual mass growth *a priori*, the mass margin should match a forecast of the anticipated mass growth. These ideas lead to observation 2-B:

2-B. In order for a novel program to be successful, the amount of mass growth must be successfully forecast so as to avoid a gross overallocation or underallocation of design margin.

These observations of the necessity of a successful forecast motivate the research objective of this thesis.

Research Objective: to create a methodology for estimating the overall system technical performance uncertainty through the forecasting of potential performance shortfalls for a novel concept such as a reusable booster system

In order to further examine these forecasts, chapter 2 will discuss the nature of uncertainty and the sources of uncertainty that directly affect a novel development program. Chapter 3 will document the current ways that margin is estimated across different industries and disciplines. Chapter 4 will outline a proposed methodology for forecasting mass growth and determining design margin, and Chapter 5 will outline the example problem to be used to examine this methodology.

CHAPTER II

UNCERTAINTIES IN SPACE AND LAUNCH VEHICLES

Chapter 1 established that while space and launch vehicles will experience mass growth and performance shortfalls, the degree of this mass growth and performance shortfall is subject to uncertainty. Mass margins as established in Chapter 1 are intended to mitigate the technical and programmatic risk manifested by this uncertainty. Before evaluating methods of estimating design margin, the nature of the underlying uncertainty must be explored.

The word uncertainty has evolved to be a common term for a large number of concepts. One common use of the term uncertainty is when referring to games of chance and similar situations which involve calculable odds. Additionally, the concept of uncertainty can apply to situations of incomplete knowledge. These two distinct concepts, both of which are forms of uncertainty, are known as aleatory and epistemic uncertainty respectively. [132]

Uncertainty due to the inherent randomness of a physical system or the environment is known as aleatory uncertainty. [50] This type of uncertainty is traditionally represented as probability distributions applied to uncertain variables. Aleatory uncertainty is also known as randomness, stochastic uncertainty, and irreducible uncertainty. [152, 30] Examples of this type of uncertainty include weather patterns, games of chance, and manufacturing variability.

Uncertainty due to one's lack-of-knowledge of the state of a system is known as epistemic uncertainty. This includes uncertainty due to measurement devices, lack of data, and analysis model assumptions. [108, 152, 99] This term is derived from Greek word for knowledge, *episteme*. [132] This type of uncertainty is also known as reducible uncertainty. As knowledge of the underlying system increases, the epistemic uncertainty will be reduced. The distinction is made from aleatory or irreducible uncertainty because epistemic uncertainty is theoretically reducible. [30]

This chapter examines these uncertainties from two perspectives. The first section approaches uncertainty from a development program perspective. Next, uncertainties in other engineering and scientific disciplines are examined. Ideas from these disciplines are then synthesized into a taxonomy of uncertainty specific to the development of space and launch vehicles.

2.1 Endogenous and Exogenous Sources of Uncertainty

The initial approach to the uncertainties affecting the development of a space or launch vehicle is from a program office-centric view. Previous work into categorizing cost and weight growth has made a distinction between factors within a project's control and factors outside of the project's control. [44, 85, 130, 151] A similar distinction will be made with regard to uncertainties affecting development programs. Endogenous sources of uncertainties are uncertainties which can be traced to sources within a program office's control while external sources of uncertainty can be traced to sources outside of a project's control.

2.1.1 Endogenous Sources of Uncertainty

Endogenous sources of uncertainty can be traced to within the development program office. Three factors traditionally account for endogenous uncertainties—assumptions about the performance of new technologies, optimistic assumptions during early design, and the omission of subsystems during early design. [130]

In order to meet more demanding performance objectives, programs often incorporate new technologies into space and launch vehicles. Programs are often initiated while technologies are still in development leading to assumptions of technology impacts being made during early design. These assumptions are often optimistic leading to eventual performance shortfalls when the optimistic predictions fail to materialize. Examples of programs that experienced significant technological uncertainty include the X-33 and the National Aerospace Plane. [130]

A larger endogenous uncertainty can be attributed to the immaturity of the design during conceptual design. During early design, initial estimations and assumptions have to be made in order to begin the iterative cycles of design. Furthermore, as the vehicle has

not been fully defined yet, analyses during early design are often limited to lower-fidelity codes. The history of vehicle development has shown that these initial estimations and assumptions are often optimistic. As the design progresses, the initial estimations and assumptions are replaced with more detailed designs. Additionally, heritage components or subsystems slated for use with a new design can also be found to be inadequate requiring new subsystem designs. Additionally, while less significant than other design uncertainties, some subsystems might be omitted from initial designs. [130]

In addition to the gradual growth due to the increased definition of design, development programs will encounter technical problems which may require a component or subsystem redesign. Any potential need for a component or subsystem redesign introduces additional uncertainty into a subsystem or vehicle. Examples of vehicle subsystems in recent history which required redesigns due to technical problems include the Boeing 787 composite wing-box and the Ares I Upper Stage which required redesign due to thrust oscillation. [83, 11]

2.1.2 Exogenous Sources of Uncertainty

Sources of uncertainty whose origins can be traced to outside of a program development office can be classified as exogenous uncertainties. Primarily this refers to uncertainties in requirements, but this can also refer to externally mandated changes due to interfacing with other aspects of an integrated architecture. [130]

The largest source of exogenous uncertainties appears in the form of requirements uncertainty. While a program office can work with the customer to influence requirements, ultimately, the responsibility for requirements changes lies outside of the program office. [130] Requirements changes can come in two flavors: through changes in scope as well as changes in constraints. Changes in scope refer to increasing demands on the system under design beyond the original proposal. Constraints refer to requirements which limit the design space; an example of a design change due to a constraint is a change in the Space Shuttle External Tank's foam insulation formula which was required to no longer contain freon. [101]

Another source of exogenous uncertainties stems from interactions within an architecture

or system. In the case of a vehicle's development, changes may be mandated so that the vehicle can work within a larger architecture; changes may also be mandated so that one vehicle can overcome shortcomings of other parts of an architecture in development. Examples of this type of uncertainty can be found in the Apollo lunar architecture. [130] If one is looking at integrating a subsystem with a larger vehicle, externally mandated changes can apply to subsystems to fix integration issues. Examples of this can be found in Section 1.2.1.4 where rocket engine masses had to grow to account for stage integration issues.

2.1.3 The Effects of Both Endogenous and Exogenous Uncertainties

Both endogenous and exogenous sources of uncertainty can have significant impacts on a development program. Documenting these effects is problematic as most past vehicle programs only document bulk mass numbers—differentiating between internally versus externally motivated growth is impossible. [130]

One program which tracked the sources of weight growth was Skylab, the United States' first space station. Figure 7 shows the mass growth of the program due to both the design maturation (endogenous sources) as well as mission requirements (exogenous sources). From this figure, the spacecraft suffered approximately 17% mass growth due to mission requirements and 23% mass growth due to design maturation for a total of 40% mass growth. [54]

While it is impossible to specify the relative magnitude of endogenous versus exogenous uncertainties *a priori*, the Skylab case study shows that it is possible for the effects of both sources to be on the same order of magnitude. Neither source can be neglected in an analysis of uncertainties and the resulting design margin estimate.

2.2 Treatment of Uncertainties in Literature

While Section 2.1 provided a treatment of uncertainties by looking at vehicle development programs, many other scientific and engineering disciplines also have treatments of uncertainty. In this section, taxonomies of uncertainties from other disciplines will be examined.

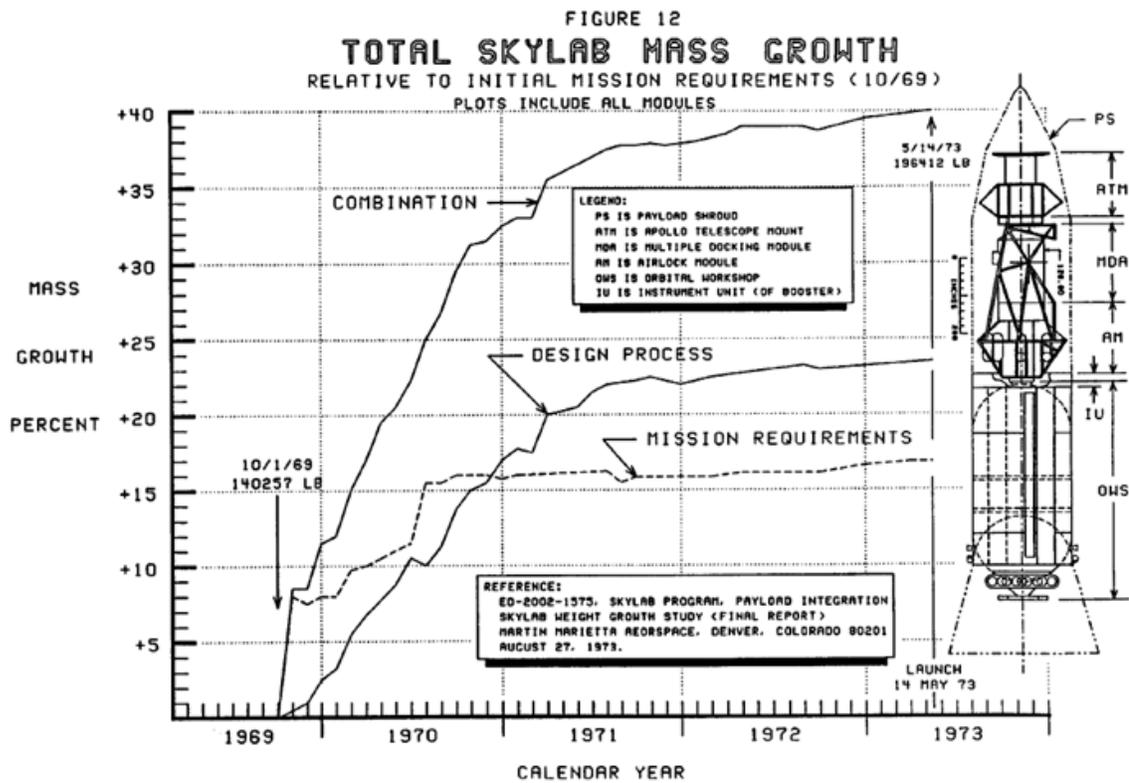


Figure 7: Dry Mass Growth of Skylab [54]

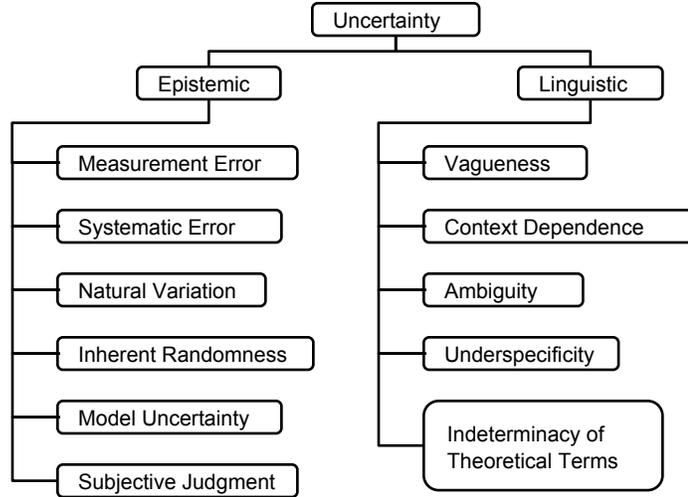


Figure 8: Taxonomy of Uncertainties in Ecology [108]

2.2.1 Ecology and Conservation Biology

In the field of Ecology and Conservation Biology, Regan et al. differentiate uncertainty into epistemic and linguistic uncertainties. Epistemic uncertainties in this classification include uncertainties due to a lack of knowledge as well as aleatory uncertainties. Linguistic uncertainties arise due to the use of natural language. The complete taxonomy of uncertainty by Regan et al. can be seen in Figure 8. [108]

2.2.1.1 *Epistemic Uncertainties*

Within this taxonomy, epistemic uncertainties are subdivided into measurement error, systematic error, natural variation, inherent randomness, model uncertainty, and subjective judgment.[108]

Measurement error refers to uncertainty due to the underlying equipment and observational techniques used for making observations. Measurements of an underlying value will vary around a mean; statistical treatments utilizing multiple measurements and confidence intervals have been developed to handle this type of uncertainty. An example of measurement uncertainty in ecology is estimating the population size of a particular species.[108]

Systematic error is formally defined as “the difference between the true value of the quantity of interest and the value to which the mean of the measurements converges as

sample sizes increase.” This source of error can stem from both the intentional judgment of a scientist or the calibration of a piece of equipment. The proscribed solution to systematic bias is to carefully examine an experimental procedure in an attempt to find biases. [108]

Within this taxonomy, natural variation and inherent randomness are types of epistemic uncertainty. While both of these uncertainties could be classified as Aleatory uncertainty, they are classified as epistemic by Regan et al. Natural variation refers to the idea that a parameter of interest is subject to change over time and space; for example the population of a species will naturally change with time. Inherent randomness refers to the idea that a system is not deterministic. Inherent randomness is only included within this taxonomy for completeness as biological systems are not expected to be truly random— randomness within biology stems from the idea of natural variation. [108]

Model uncertainty results from the abstractions used to represent biological and physical systems. First, model uncertainty arises due to the compromise between model fidelity and model complexity— only the parameters which are seen to be relevant to the problem at hand are included in the model. This selection of parameters could introduce uncertainty due to possible interactions and higher order terms. Another source of model uncertainty comes from using a mathematical relationship to model observed phenomena. For example, the rate of change of a population is modeled with a mathematical derivative, but the underlying phenomena of birth, death, and migration are not mathematical in nature. In practice, model uncertainty is impossible to eliminate and difficult to quantify. [108]

The final source of epistemic uncertainty in this taxonomy is due to subjective judgment. This occurs due to the need to make statements about a subject for which there exists little empirical evidence. This type of uncertainty is traditionally addressed by having an expert express a degree of belief. This degree of belief makes use of subjective probabilities as a way to quantify the uncertainty surrounding a parameter of interest. [108]

2.2.1.2 Linguistic Uncertainties

Within this taxonomy, linguistic uncertainties are subdivided into vagueness, context dependence, ambiguity, underspecificity, and the indeterminacy of theoretical terms.[108]

The most important type of linguistic uncertainty is vagueness. Vagueness occurs because the words used in natural language allow borderline cases. This type of uncertainty leads to several problems and can lead to problems in the management task of assigning resources. While attempts have been made to eliminate vagueness by adding new, technical words with specific meanings, vagueness still occurs leading to the conclusion that it is impossible to eliminate. While impossible to eliminate, the proscribed methods of handling this type of uncertainty involve reducing the terms to multi-dimensional measures which can be used with numerical analysis. [108]

A second type of linguistic uncertainty is context dependence; this arises when a researcher or author fails to specify the context in relation to the terms being used. The obvious solution to this type of uncertainty is to explicitly specify the context within which terms are intended. [108]

Ambiguity occurs due to the fact that a word can have several meanings. Another necessary condition for ambiguity is that, within the context of the word's usage, the intended meaning is not clear. Ambiguity is occasionally confused with vagueness, but this taxonomy holds that vagueness only applies to borderline cases. The proscribed solution to ambiguity is to explicitly specify the intended meaning. [108]

Another source of linguistic uncertainty is due to underspecificity or unwanted generality. This occurs when statements fail to provide the desired level of specificity. Again, the term "vague" has been used to describe things that are underspecific, but this taxonomy seeks to differentiate vagueness as due to borderline cases. [108]

The final source of linguistic uncertainty is due to the indeterminacy of theoretical terms. One source of this uncertainty is due to the potential future definitions of current technical terms—potential future ambiguities. Another source of this uncertainty comes from the fact that definitions of certain terms are not agreed upon by the relevant communities. This situation typically occurs in emerging fields which have not had the necessary time to settle definitions of theoretical terms. [108]

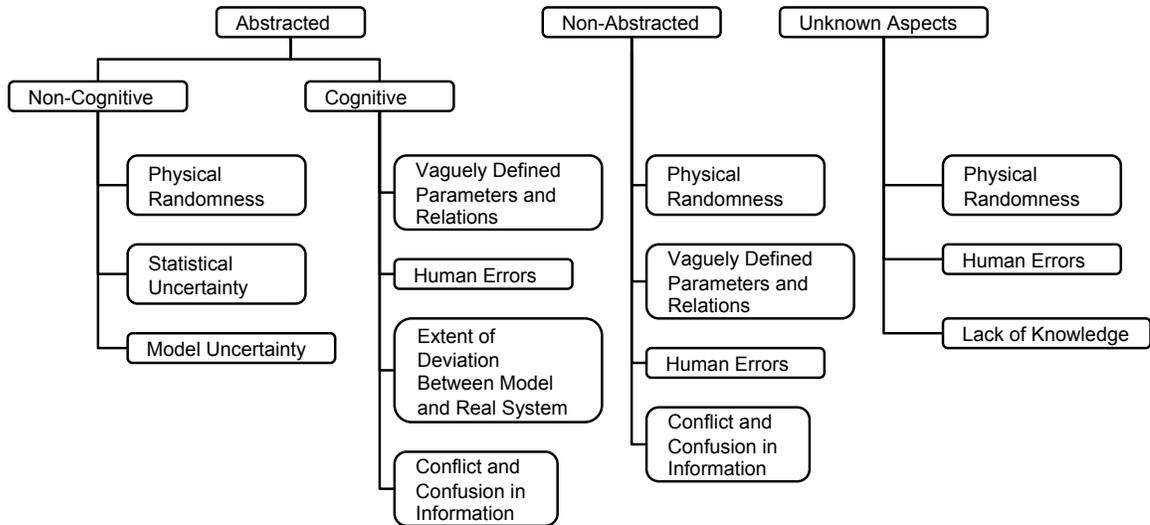


Figure 9: Taxonomy of Uncertainties in Civil Engineering [7]

2.2.2 Civil Engineering

In their chapter, “Uncertainty Modeling in Civil Engineering with Structural and Reliability Applications,” Ayyub and Chao specify a taxonomy of uncertainty for civil engineering. Their taxonomy focuses on the idea that a model of a system drives uncertainty. As can be seen in Figure 9, this taxonomy is broken down into three primary classifications of uncertainty: uncertainties due to model abstractions, uncertainties independent of a model’s abstraction, and unknown unknowns. [7]

Abstracted uncertainties stem from the fact that engineers conduct analyses on a model of a real system— an abstraction of reality. This category of uncertainty can be further subdivided into non-cognitive and cognitive sources of abstracted uncertainty. Cognitive, abstracted uncertainties “arise from mind-based abstractions of reality;” this type of uncertainty can be subdivided into human errors, vaguely defined parameters and relations, deviation between a model and a real system, and conflict and confusion in information. Because these uncertainties pertain to the act of creating a model, probability and statistics cannot properly model the effects of these uncertainties. In contrast, non-cognitive, abstracted uncertainties are primarily modeled by probability distributions. This type of uncertainty can be subdivided into physical randomness, statistical uncertainty, and model uncertainty. [7]

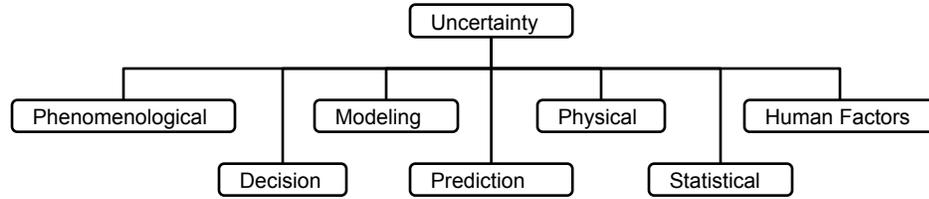


Figure 10: Taxonomy of Uncertainties in Structural Engineering [84] adapted from [132]

Non-Abstracted uncertainties are uncertainties due to the assumptions made when deciding which aspects of a system do not need to be abstracted into a model. Traditionally the distinction between abstracted and non-abstracted uncertainties is due to the choices made by analysts when determining the fidelity level required of a model. These uncertainties include physical randomness, vaguely defined parameters and relations, human errors, and conflict and confusion in information. These sources of uncertainty are much more difficult than abstracted uncertainties to quantify because they arise from the ability of a model to mimic a real system. [7]

The final type of uncertainty identified in this taxonomy is due to the unknown aspects of a system and refers to aspects of reality not accounted for by models. These unknowns can manifest themselves through three mechanisms: physical randomness, human errors, or a lack of knowledge. While human errors appear in other portions of this taxonomy, within this branch, human errors refer to failure modes or aspects of a model intentionally left out of a system. Similarly, physical randomness in this branch refers to randomness not accounted for by the model. Finally, a lack of knowledge is fundamental to this branch of uncertainty as incomplete knowledge can lead to substantial unknowns in a development project. [7]

2.2.3 Structural Engineering

In his book, *Structural Reliability Analysis and Prediction*, Robert Melchers develops a taxonomy of uncertainty for the field of structural engineering. This taxonomy can be seen in Figure 10. [84]

The first main type of uncertainty in this taxonomy is phenomenological uncertainty. This form of uncertainty refers to 'unimaginable' phenomenon which can cause failures.

Melchers defines phenomenological uncertainty as arising “whenever the form of construction or the design technique generates uncertainty about any aspect of the possible behaviour of the structure under construction, service and extreme conditions.” This form of uncertainty is of particular importance when projects seek to advance the state of the art. [84]

The second form of uncertainty within this taxonomy is decision uncertainty. The decision in question within this taxonomy is that of whether a limit state violation has occurred. Typically these decisions involve questions of crack widths and deflections. Furthermore, this type of uncertainty can be expressed as a relative loss of structural usefulness. [84]

Modeling uncertainty is the uncertainty associated with using a simplified relationship (a model) to represent the ‘real’ phenomenon of interest. Typically this uncertainty has to do with a lack of knowledge and will be reduced through research or the increased availability of data. [84]

The fourth type of uncertainty within this taxonomy is that of prediction uncertainty. This uncertainty refers to the accuracy of a prediction of a future state given the present state of information. As development progresses, new information will allow better estimations and predictions— new information becomes known up through construction and operation which allow for the best possible reliability predictions. [84]

Physical uncertainty within this taxonomy is concerned with the inherent random nature of certain variables. While some physical uncertainty can be reduced through greater knowledge, most sources of physical uncertainty cannot be reduced. Typically this uncertainty is not known *a priori* and must be empirically assessed or estimated subjectively. [84]

Statistical uncertainty stems from using sample statistical estimators to represent a random variable with a probability density function. The observations used to construct these estimators usually contain some degree of bias, and this bias leads to uncertainty. This uncertainty can be treated in a reliability analysis by assigning ranges to the parameters which describe the probability density function. [84]

The final form of uncertainty considered within this taxonomy is that of uncertainties due

to human factors. Melchers subdivides this category into uncertainties due to human error and uncertainties due to human intervention. Human error is specifically included in this taxonomy due to the historical cases of human errors leading to structural failure. Human errors can further be subdivided into errors which occur during accepted engineering and maintenance practices and those which occur due to ignorance or due to a lack of adherence to established procedures. Human intervention refers to the actions of humans during the design, documentation, construction, and usage of structures. These interventions can be institutional through standardized inspections or informal resulting from an observation that 'something is wrong.' [84]

2.2.4 Systems Engineering

Within the field of Systems Engineering, uncertainty is traditionally looked upon in terms of risk. The definition of risk in the NASA Systems Engineering Handbook has two parts: the probability of an undesired event and the consequences of the undesired event. The first part of this definition deals with uncertainty therefore the systems engineering taxonomy of risk is also a taxonomy of uncertainty. [92, 132]

Uncertainties within systems engineering are broken down into technical, schedule, cost, and programmatic uncertainties. Technical is uncertainty associated with the ability of the system to meet technical requirements and meet stakeholder expectations; these uncertainties stem from the evolution of the design of a program. Schedule uncertainties are associated with the adequacy of the time allocated for the completion of a task or tasks. Cost uncertainties refer to both the funding allocated to program development as well as the ability of the system under development to meet cost objectives. Finally, programmatic uncertainties are uncertainties which originate from outside of the development office which are realized as impacts on technical performance, cost, and schedule. [92]

2.2.5 Modeling and Simulation

Oberkampf et al. seek to differentiate between traditional uncertainties and error in their taxonomy of uncertainty. This taxonomy, seen in Figure 11, characterizes uncertainties as either being aleatory, epistemic, or being due to error in the modeling and simulation. [99]

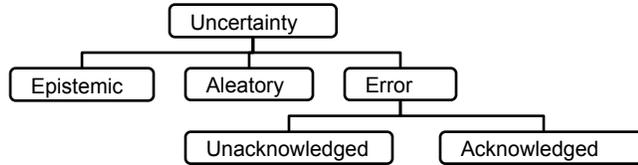


Figure 11: Taxonomy of Uncertainties in Modeling and Simulation [99]

The classifications of epistemic and aleatory uncertainty in this taxonomy are consistent with the definitions provided in the introduction to this chapter. Aleatory uncertainty refers to the inherent variation of the subject under consideration. Epistemic uncertainty is defined as “a potential inaccuracy in any phase or activity of the modeling process that is due to a lack of knowledge.” [99]

The key distinction in this taxonomy is the use of the term ‘error.’ Error is defined “as a recognizable inaccuracy in any phase or activity of modeling and simulation that is not due to lack of knowledge.” Errors are further broken down into acknowledged and unacknowledged errors. Acknowledged errors are errors that an engineer or analyst has an idea of the relative impact of errors; for example, a lack of precision in a floating point number is an acknowledged error. Conversely, unacknowledged errors are inaccuracies which are not recognized by an analyst or engineer but which are recognizable. [99]

2.2.6 Space Architectures and Systems

As part of his PhD thesis, Myles Walton introduced a taxonomy of uncertainty as applied to space architectures and systems. This taxonomy acts as a lens to focus on the stages of a space system’s lifecycle. Uncertainties are broken down into development, operational, and model uncertainties. This breakdown was chosen to highlight the operational uncertainties inherent in a space system. The entirety of the taxonomy can be seen in Table 8. [142]

2.2.7 Design of Complex Systems

As part of his PhD thesis, Daniel Thunnissen created a taxonomy of uncertainties as applied to the design of complex systems. As his thesis examines uncertainties surrounding system development and later establishes a methodology for estimating a design margin,

Table 8: Taxonomy of Uncertainties for Space Architectures and Systems [142]

Development Uncertainty	Operational Uncertainty
Political Uncertainty– uncertainty of development funding instability	Political Uncertainty– uncertainty of operational funding instability
Requirements Uncertainty– uncertainty of requirements stability	Lifetime Uncertainty– uncertainty of performing to requirements in a given lifetime
Development Cost Uncertainty– uncertainty of developing within a given budget	Obsolescence Uncertainty– uncertainty of performing to evolving expectation in a given lifetime
Development Schedule Uncertainty– uncertainty of developing within a given schedule profile	Integration Uncertainty– uncertainty of operating within other necessary systems
Development Technology Uncertainty– uncertainty of technology to provide performance benefits	Operations Cost Uncertainty– uncertainty of meeting operations cost targets
	Market Uncertainty– uncertainty in meeting demands of an unknown market
Model Uncertainty	

his taxonomy of uncertainty can provide insight into the nature of uncertainties in space and launch vehicles. This taxonomy, pictured in Figure 12, features four primary types of uncertainty: ambiguity, epistemic uncertainty, aleatory uncertainty, and interactions. [132]

The first main type of uncertainty in this taxonomy is ambiguity where uncertainty results from the use of imprecise terms. Ambiguity is also known as imprecision, design imprecision, linguistic imprecision, and vagueness. This type of uncertainty can be reduced through linguistic conventions and careful definitions, but it will always be present due to the nature of human discourse. [132] This classification is similar to linguistic uncertainty which is more fully developed by Regan et al.; a summary of Regan et al. can be found in Section 2.2.1.2.

Another fundamental uncertainty is aleatory uncertainty. The definition of aleatory uncertainty within this taxonomy is consistent with the definition presented in the introduction of this chapter. Thunnissen defines aleatory uncertainty as “the inherent variation associated with a physical system or environment.” [132]

A third primary type of uncertainty within this taxonomy is uncertainty due to interactions. This uncertainty arises due to potential unanticipated interactions of events, systems, and/or disciplines. These interactions are, in theory, foreseeable, but the complex nature

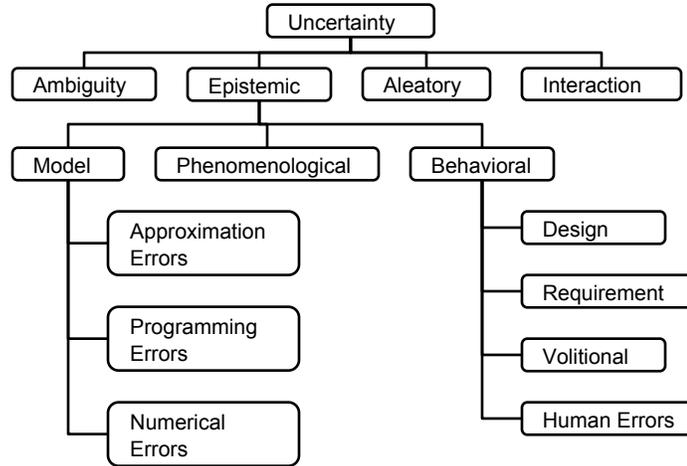


Figure 12: Taxonomy of Uncertainties in the Design of Complex Systems [132]

of the problem makes these uncertainties difficult to predict. [132]

2.2.7.1 *Epistemic Uncertainties*

The largest set of uncertainties in this taxonomy falls under epistemic uncertainty. Thunnissen’s definition of epistemic uncertainty is similar to the definition from earlier in the chapter— he defines epistemic uncertainty as “any lack of knowledge or information in any phase or activity of the modeling process.” He further breaks epistemic uncertainty down into model, phenomenological, and behavioral uncertainty. [132]

Model uncertainty is defined as “the accuracy of a mathematical model to describe an actual physical system of interest.” This type of uncertainty falls under the category of epistemic uncertainty because models are simplifications of reality used to represent actual phenomena. Model uncertainty can be further broken up into approximation, numerical, and programming errors. Approximation errors refer to the use of lower-fidelity tools to represent reality; this uncertainty can be reduced by using higher fidelity tools. Numerical errors occur because a computer system cannot represent real numbers, only a floating point approximation. Programming errors are blunders in a piece of code. [132]

The second category of epistemic uncertainty is phenomenological uncertainty. This category covers uncertainties which can be classified as “unknown unknowns.” These unknowns traditionally refer to information that cannot be known at the time of making

decisions. Furthermore, these uncertainties are particularly important when considering the development of novel concepts. [132]

The third subcategory of epistemic uncertainty is behavioral uncertainty. This category encompasses uncertainties which result from the actions or organizations or individuals. This category is further subdivided into design uncertainty, requirement uncertainty, volitional uncertainty, and human errors. Design uncertainty occurs because designers have a design decision which has not yet been decided. Requirements uncertainty occurs when stakeholders independent of the program office change requirements. Volitional uncertainty is uncertainty about what individuals or program offices at any level of a contractual hierarchy will decide in the future; within this taxonomy volitional uncertainty is separated from design uncertainty as it refers to the decisions that entities make when performing non-design tasks such as estimating performance. Finally, human errors are blunders by the program's staff. [132]

2.3 Taxonomy of Uncertainty

Section 2.2 explored taxonomies of uncertainties in scientific and engineering fields. In each of these fields, the specific taxonomy of uncertainty acted as a lens to focus on the uncertainties most pertinent to the respective field. In this case, a taxonomy that focuses on the development of space and launch vehicles is needed. This taxonomy should capture the relevant sources of uncertainty which affect mass growth and performance.

The complete taxonomy can be seen in Figure 13. The first distinction of uncertainties is between aleatory and epistemic uncertainty. Aleatory uncertainty (also known as irreducible uncertainty) is defined as the inherent randomness of a physical system or environment. [50] This uncertainty will only manifest itself during manufacturing or operation. For example, the specific dimensions of a manufactured part or the day-of-launch wind profile is subject to uncertainty. Because these uncertainties only manifest themselves during the later stages of a system's lifecycle, aleatory uncertainties contribute very little to the total uncertainty encountered during a development program.

Epistemic, or reducible, uncertainty is due to a lack of knowledge. [108, 152, 99] As can

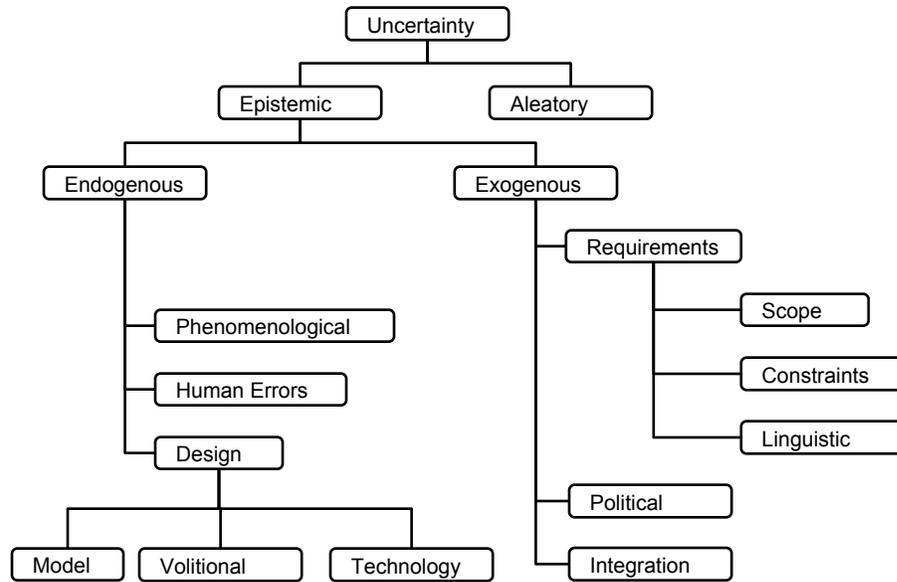


Figure 13: Taxonomy of Uncertainties in the Development of Space and Launch Vehicles

be seen by Thunnissen’s taxonomy in Figure 12, types of epistemic uncertainties dominate the uncertainty of complex systems. [132] Similarly, epistemic uncertainties are dominant for space and launch vehicles. Examples of epistemic uncertainties can include the assumptions used in the models used to initially size a vehicle, assumptions about new technologies used in a vehicle, or the requirements that will ultimately dictate the size of the vehicle.

As epistemic uncertainty dominates the uncertainty of space and launch vehicles, the key aspect of this taxonomy is how it differentiates between different epistemic uncertainties. This taxonomy focuses on how uncertainty affects a program from the point of view of a program office because the program office is responsible for conducting risk management and mitigation programs as well as being ultimately responsible for determining the amount of design margin carried in a design. Risk management programs have an “ownership” phase in order to distinguish which risks and associated responses are managed by the program office and which are managed by other organizations. [25] Based on this need to distinguish ownership, epistemic uncertainties can be subdivided into endogenous and exogenous uncertainties. Endogenous uncertainties are uncertainties whose origin can be traced to within the authority of the program office; exogenous uncertainties are uncertainties whose origin can be traced to outside of the program office.

This distinction can be used on all levels of a program’s contractor tree. Subcontractors trying to identify epistemic uncertainties can use this taxonomy as a guide to distinguish between factors that they are responsible for and factors under the prime contractor’s control which will impact the subcontractor. Similarly, a prime contractor can distinguish between uncertainties under its control and factors which the customer or government is responsible for.

2.3.1 Endogenous Uncertainties

Endogenous uncertainties arise from within the authority of a program office. As these are epistemic uncertainties, the program office can invest resources in reducing these uncertainties as part of a risk mitigation program or as a natural occurrence during the progression of the design. Endogenous uncertainties can be further divided into phenomenological uncertainties, human errors, and design uncertainties.

2.3.1.1 Phenomenological Uncertainty

A key type of uncertainty when dealing with novel concepts is phenomenological uncertainty. [132, 109] Within this taxonomy, phenomenological uncertainty is defined as uncertainty due to an acute lack of knowledge of phenomena, physical or otherwise; this lack of knowledge is characterized by the inability to recognize potential risks or opportunities. Specific phenomenological uncertainties are also referred to as “unknown unknowns.”

Phenomenological uncertainty is central to the development of novel concepts featuring fundamentally new designs and concepts of operation because a goal of these programs is to reduce uncertainty for future development programs. These development programs will have to make design decisions under uncertainty because relevant knowledge of the system cannot be known, even in principle, at the time of making decisions. [132]

History has numerous examples of phenomenological uncertainties in novel concepts. The first U.S. satellite, Explorer I, began to nutate away from its axis of rotation due to unimagined physical phenomena.

2.3.1.2 Human Errors

Another source of uncertainty in the development of space and launch vehicles is human error. Human errors are defined as blunders occurred in the design, manufacture, test, or operation of a system. Errors can lead to corrective redesigns, manufacturing rework, or the loss or partial loss of parts during operation. This uncertainty can be reduced by active programs to catch errors quickly.

2.3.1.3 Design Uncertainties

A large subcategory of endogenous uncertainty is made up of design uncertainties. Design uncertainties can be traced to a lack of knowledge in the design of the system in question. This lack of knowledge has two forms: uncertainty due to assumptions made during early design activities and uncertainty due to yet-to-be-determined design decisions. Design uncertainties can further be decomposed into model uncertainties, volitional uncertainties, and technological uncertainties.

The first form of design uncertainty is model uncertainty. Traditionally, this term is used to refer to epistemic uncertainties resulting from the fidelity level of the analysis tools used to conduct analyses. [84, 108, 132] Within this taxonomy, model uncertainty not only refers to the fidelity level of analysis tools, but also the fidelity level of the engineers' and designers' mental model of the system under development. This mental model contains numerous assumptions about the nature of the vehicle manifest themselves as inputs of analysis tools. Because the fidelity of analysis tools is directly related to the fidelity of the inputs (the garbage in, garbage out phenomena), uncertainties due to the overall fidelity level of the design is categorized as model uncertainty.

Another form of design uncertainty related to assumptions made during early design is technological uncertainty: uncertainty stemming from the incorporation of new technologies in a system. These uncertainties can take different forms throughout the development cycle of a concept. The most common form of this uncertainty is through unrealized performance gains or cost reductions when optimistic projections fall short. Another form of uncertainty

is where new technologies fail to work and subsystems must be replaced with more conventional alternatives as was the case with the composite LH2 tank on the X-33. Finally, new technologies may work in terms of technical performance while requiring additional resources to maintain during operations e.g. the Orbiter's thermal protection system.

The third form of design uncertainty within this taxonomy is volitional uncertainty: uncertainty due to the decisions of actors within the design process of the system. Within this taxonomy, this primarily refers to future design decisions which can manifest itself in two ways. The first is through future design decisions which will add detail to a low-fidelity or low-resolution design; while these decisions usually add detail which results in a higher-fidelity vehicle within the original design assumptions, occasionally these decisions add new subsystems or subsystems which were omitted during earlier phases of design. The second form is through decisions which fundamentally change the design of the system in question. These changes can be in response to decreasing technical performance or in response to the results of higher-fidelity analyses. A historical example of fundamental changes over the development of a system can be seen with the Orbiter which experienced changes in its wing planform, overall length, and provisions for air breathing engines. [61]

2.3.2 Exogenous Uncertainties

Exogenous uncertainties are uncertainties whose origin can be traced to outside of the influence of the development program office. While these uncertainties will be reduced throughout the course of program execution, unlike endogenous uncertainties, the program office cannot expressly conduct risk reduction programs to explore this uncertainty. Exogenous uncertainties can be subdivided into requirements uncertainty, political uncertainty, and integration uncertainty.

2.3.2.1 Requirements Uncertainties

A large source of exogenous uncertainty comes in the form of requirements uncertainty. While program offices are able to give input into requirements at nearly all phases of the design, requirements uncertainty is classified as exogenous because the ultimate authority in specifying requirements lies with a customer or government. Requirements uncertainty

is defined as uncertainty due to the ambiguity of specified requirements or due to an unexpected change in customer requirements. Requirements uncertainty can take three forms: scope uncertainty, constraint uncertainty, and linguistic uncertainty.

The first form of requirements uncertainty has to do with the scope of the project. Many programs experience scope creep— an increase in the demanded technical performance of a system over the originally agreed upon requirements. [73] This form of uncertainty deals directly with the technical performance requirements of a system, otherwise known as functional requirements. [27]

The second type of requirements uncertainty is associated with the constraints placed on the system under development. Unlike uncertainty associated with scope, constraint uncertainty does not deal with technical performance requirements, but instead deals with non-functional system requirements, requirements which are attributes or constraints on a system. [27] An example of a change in a constraint for a program is from the Space Shuttle Program— the External Tank’s foam insulation was constrained to no longer contain freon. [101] Additionally, non-functional requirements can specify safety, reliability, maintainability, etc. thresholds which can further constrain development.

A third type of requirements uncertainty is due to the natural language used to specify requirements. Regan et al. in their taxonomy of uncertainties relating to ecology and conservation biology recognized that linguistic uncertainty contributes to the overall uncertainty in situations which involve policy and decision making. [108] In the context of space and launch vehicle development, uncertainty in language will appear in the requirements specification. This uncertainty can lead to misunderstandings between a program office and a customer or government; these misunderstandings can result in wasted effort and schedule slippage while the contractor and customer work to understand one another.

2.3.2.2 Political Uncertainties

In his taxonomy of uncertainty for space architectures and systems, Walton introduced the concept of political uncertainty defined as the uncertainty of development funding instability. [142] This concept is incorporated in this taxonomy of uncertainty as an exogenous

source of uncertainty. Political uncertainty can have far reaching effects into vehicle development both through design decisions and the effort expended on a program.

Programs facing funding uncertainty may choose to mitigate this risk by making design decisions which are lower-cost in the short term at the expense of lifecycle costs; this can result in the omission of promising technologies in exchange for proven systems. An example of this is the decision for the Space Shuttle stack to use solid rocket motors instead of liquid boosters because of lower development cost of solid rockets. [61]

Another manifestation of political uncertainty is in the level of effort able to be put into vehicle development. If a company is not assured of future funding, it will not actively hire or reassign additional workers to a development project in anticipation of a funding reduction. Even if a funding reduction never occurs, the uncertainty associated with a possible funding reduction will increase the cost and create a schedule slippage as necessary additional workers cannot immediately work on a project.

2.3.2.3 Integration Uncertainties

A third exogenous uncertainty stems from issues inherent in integrating a system or subsystem within a larger system, architecture, or system of systems. This uncertainty stems from the idea that individual projects or subsystems within a larger system or architecture will develop at different rates; in some cases, such as with propulsion systems, subsystem development can occur independently of vehicle or architecture development. This asynchronicity in development leads to integration issues where the subsystem with more design freedom will be required to make changes to accommodate the more mature system. An example of this would be a launch vehicle's thrust structure being changed to accommodate the loads of the propulsion system. Another instance where integration issues introduce uncertainty is through asynchronous weight growth and performance losses where one subsystem will have to make design changes to accommodate performance losses in other parts of the system. An example of this occurred during the development of the Saturn V rocket where the S-II stage had to make design and manufacturing process decisions to minimize weight because the S-IV B stage was already in production. [14]

Integration uncertainty is the same as a design uncertainty, but the difference is in the level and contractual relationships of the program which is conducting an analysis. For instance, the integration issues between a propulsion system and a stage represent the same uncertainty. If the relationship between the propulsion contractor and stage integrator is that of a prime contractor and subcontractor which is designing an engine to prime contractor requirements, then integration issues represent a design uncertainty (subclass of model uncertainty) to the prime contractor and an integration uncertainty to the propulsion subcontractor. Another contractual relationship may give the propulsion system subcontractor more independence; for example, the subcontractor may design the same engine for use in multiple stages. In this case, integration issues will be an exogenous, integration uncertainty for both parties.

2.4 Uncertainty Elicitation

As the taxonomy presented in Section 2.3 is intended to guide organizations in identifying uncertainties in programs, the next step in an uncertainty identification and quantification exercise is to elicit forecasts for the impacts of uncertainties. These uncertainty forecasts are then used to drive a design margin estimation. Based on this taxonomy of uncertainty, three families of elicitation are used to forecast uncertainties: traditional engineering analysis, programmatic risk and opportunity analysis, and technical risk and opportunity analysis.

2.4.1 Traditional Engineering Analyses

For the purposes of this classification, traditional engineering analysis refers to the physics-based analysis of a vehicle. This set of analyses focuses on answering the question, “given a baseline vehicle conceptual design, what are its ultimate characteristics?” The most important part of this question is the fact that it assumes a baseline—traditional engineering analyses can only answer questions about how a design will evolve *assuming no fundamental changes to a design*. This form of analysis can address aleatory, model, and some technological uncertainties.

Aleatory uncertainties manifest themselves during the manufacturing and operation of space and launch vehicles; it is the role of traditional engineering analyses to determine

the likely impacts of this uncertainty. These impacts are then incorporated into the design through two mechanisms: updated weight estimations and performance margin. In this case, updated weight estimations will account for the stochastic nature of manufacturing accounted for by tolerances on part designs. As this occurs during detailed design, manufacturing uncertainties have little to no impact in the determination of design margin. A performance margin is used to account for aleatory uncertainties in the operation of a vehicle—for example, a launch vehicle will need ample performance to account for day-of-launch winds and other trajectory dispersions. The performance margin is a requirement and is independent of the design margin.

Model uncertainties are defined as uncertainties due to a lack of design fidelity and are, by definition, subject to traditional engineering analyses as a way to estimate uncertainty. These analyses focus on bounding the model uncertainty, and these bounds are then used to help determine an appropriate design margin. Similarly, technological uncertainties can be examined in the same fashion where the engineering analysis is used to bound the potential of a new technology. However, engineering analysis is only applicable for forecasting bounds of a technology to be used; it fails in the event that the technology is not included in the eventual production version of the vehicle.

2.4.2 Programmatic Risk and Opportunity Analyses

The second fundamental method of eliciting uncertainty forecasts has to do with looking outside of the development effort to characterize exogenous uncertainties. This class of problems is addressed with a programmatic risk and opportunity analysis. This type of analysis is fundamentally different from traditional engineering analysis and addresses both technical and non-technical issues which affect the vehicle. Fundamentally, this analysis looks for possible changes in the vehicle baseline due to externally motivated factors. This analysis can cover all types of exogenous uncertainties: requirement, political, and integration.

2.4.3 Technical Risk and Opportunity Analyses

Technical risk and opportunity analysis is similar in nature to programmatic risk and opportunity analysis, but the focus of technical risk and opportunity analysis lies in characterizing

endogenous uncertainties not covered by traditional engineering analyses. The goal of this analysis is to identify potential vehicle baseline changes motivated by technical reasons. This analysis covers phenomenological uncertainty, volitional uncertainty, human errors, and technological uncertainties which result in removing the technology in question from the vehicle baseline.

2.4.4 Observation

As was noted in Observation 2-B, the success of many novel concepts can be traced to the successful forecast of a margin estimation. The subsequent taxonomy of uncertainty delineated the underlying uncertainties which need to be addressed by a margin forecast, and these sources of uncertainty were further grouped into three distinct families of forecasting problem. This progression leads to Observation 3:

3. In order to fully address the underlying uncertainties relevant to a space or launch vehicle, a margin estimation technique must include forecasts for mass growth due to increased definition of an existing design as well as forecasts for potential design changes motivated by endogenous and exogenous sources of uncertainty.

This observation will guide the examination of the state of the art methodologies for estimating margin across different industries. This examination is the subject of Chapter 3.

CHAPTER III

METHODS OF ESTIMATING DESIGN MARGIN

As outlined in Chapter 1, launch and space vehicle designers address uncertainty in performance through the use of a design margin. However, this practice is not limited to the space and launch vehicle industry— the concept of a design margin is applied in several fields.

The shipbuilding industry also rigorously tracks mass and accounts for mass growth with a design margin. For surface ships, weight engineers track both the total weight and the vertical moment; for submarines, the total equilibrium of the ship is tracked. For each of these dimensions of mass properties, a design margin can be assigned. Unlike launch and space vehicle design margins, the margins for naval vessels also need to take mid-life refits into account adding another dimension of uncertainty to the problem. Finally, for the case of submarines, the design margin physically manifests itself as ballast— any mid-life refits must be offset by a change in the ballast so that the boat maintains equilibrium. [19, 122, 126]

Cost engineering is another discipline in which the use of margins is ubiquitous. Within this community, a margin is applied to a project's estimated cost to make up for likely cost over-runs; costs in a project tend to increase over time just as the mass of a space or launch vehicle increases over the design cycle. Because construction projects occur more often than the development of a space or launch vehicle, this community has far greater experience in different methods used to estimate margin. From this large experience base, this community has been able to conduct studies into the relative accuracy of various methods of margin estimation. [1, 9, 21, 59, 80]

A third discipline which utilizes the concept of margin is the estimation of project schedules. [37] Schedules are more complicated than cost or mass estimates because of the direct dependencies between activities— tree structures must be used to represent elements of a project to identify possible critical paths. Additionally, margin must be assigned to

individual project elements so that lead items can be ordered in anticipation of a start date. [8]

When looking at the problem of estimating the design margin for a novel concept, it is necessary to look at the methods used for estimating margin across these communities. Across all of these communities four general categories of estimating design margin have been identified: predetermined percentage, historical regression, expert opinion, and simulation analysis. [1, 60, 59, 21] These four methods can be used stand-alone or as hybrid methods by combining different aspects of each, but hybrid methods will inherit both the advantages and disadvantages of the parent techniques. The remainder of this chapter will look at these four categories of methods to determine which families of methods are most promising for use with a novel concept. First, the necessary qualities of a margin estimation technique will be examined, and then each of these four methods will be formalized and examined for gaps in their ability to address the desired principles for a margin estimation technique.

3.1 Principles for Design Margin Estimating Techniques of a Novel Concept

Before one can evaluate existing design margin estimating techniques, general principles which should be followed by these techniques must be established. The cost engineering community has developed several principles with which margin estimating techniques should comply. These principles should then be examined to determine their applicability to novel concepts. Finally, any new principles for margin estimation techniques which apply to novel concepts must be identified and justified.

3.1.1 AACE Recommended Practice

The Association for the Advancement of Cost Engineering International (AACE) recommended practice lists several principles which should be followed by contingency estimating techniques: [1]

- Meet client objectives expectations and requirements
- Part of and facilitates an effective decision or risk management process

- Fit-for-use
- Starts with identifying the risk drivers with input from all appropriate parties
- Methods clearly link risk drivers and cost/schedule outcomes
- Avoid iatrogenic (self-inflicted) risks
- Employs empiricism
- Employs experience/competency
- Provides probabilistic estimating results

The first three items of this list deal with the interaction between the overall margin estimation technique and the management of the organization which is developing a new project. First of all, the estimation technique must meet the requirements of the organization; these requirements are characterized by specific data deliverables (point estimates or ranges) and resource availability (schedule and effort). The second principle is that a margin estimation technique should fit into the organization's larger risk management process. The third aspect is that the specific technique undertaken must work within the organization—organization-specific knowledge that may or may not be affect estimating must be taken into account when assessing particular estimation techniques.[1]

The next three items on this list deal with the process of estimating the risks. First, any margin estimation technique should begin with identifying risk drivers. Next, these risk drivers should be connected to potential outputs. These steps ensure that the estimation process is straightforward. The third part of this section involves avoiding self-inflicted risks— “the estimation process itself should not introduce new risks.” [1] Examples of new risks include having too many risks or cost items as well as having and overstated or understated risk estimate. [1]

The next two items on this list address particular inputs to the margin estimating technique. The first is that methods to estimate margin should employ empiricism. This can mean that it directly employs historical regressions or it can utilize lessons learned,

benchmarking, or validating against historical data. The other principle is to utilize experience/competency. A lack of experience in the estimators increases the iatrogenic risks of the project. Furthermore, experience becomes more critical to the success of the estimation if the methodology employed utilizes a smaller amount of empiricism. [1]

The final item addresses the output of the process. In order to facilitate decision making, probabilistic estimate outputs provide more information than point estimates. This additional information ensures that the consequences of the decision are more readily understood— probabilistic information will allow a decision maker to understand the likelihood of a margin being overrun. [1]

The AACE recommended practice recognizes that all four families of margin estimation techniques can be made to fit these principles. Table 14 lists each of these principles versus the families of margin estimation techniques. Within this table, each family of estimating methods satisfies most of these principles. There are two exceptions to the overall satisfaction of these principles. First, expert judgment, predetermined guidelines, and simulations analyses do not necessarily employ empiricism and an inherent feature of the method. Second, expert judgment and predetermined guidelines do not necessarily provide a probabilistic result. [1]

3.1.2 Principles Necessary for Novel Concept Margin Estimation

The principles laid out by the AACE recommended practice provide a good starting point for principles which can be used to evaluate margin estimation methods for a novel concept. However, these principles were written by the cost engineering community under the assumption that most every estimation will occur for a non-novel project. This assumption needs to be re-examined. Furthermore, because the subject of the estimation is a novel concept, new principles to which estimation methods should adhere to can be derived.

One principle which must be re-examined is the idea that an estimation method should employ empiricism. Explicit empiricism, basing estimates for a new project on similar previous projects, is not applicable due to the lack of previous projects. However, the key idea

First Principles	Classes of Contingency Estimating Methods			
	Expert Judgment	Predetermined Guidelines	Simulation Analysis*	Parametric Modeling
Meets client objectives, expectations and requirements	Whether a given method or combination of methods best meets the clients objectives, expectation or requirements must be determined prior each application			
Part of a risk and decision management process	Any method can potentially be incorporated in a process.			
Fit-for-use	Any method can potentially be made to address a variety of applications, but typically each method has strengths and weakness. Hybrid approaches can take advantage of the strengths of several methods.			
Starts with identifying risk drivers	Any method can potentially be made to start with identifying risk drivers.			
Links risk drivers and cost/schedule outcomes	Requires that expert(s) make and communicate the linkages	Linkages can be directly incorporated in the guidelines	Linkages are directly used in the <i>expected value</i> method	Linkage is inherent to this method
Avoids iatrogenic (self-inflicted) risks	Bias must be tempered, often through consensus	Care must be taken with risks not considered in the guidelines	Complexity of the method increases the need for disciplined approach	Care must be taken with risks not considered in the model
Employs empiricism	Generally requires the use of lessons learned, and/or validation or benchmarking using historical information (not an inherent feature of the method)			Explicitly addressed if regression based
Employs experience /competency	Expertise explicitly required	Expertise employed in development	Expertise employed in analysis	Expertise employed in development
Provides probabilistic estimating results	Can provide subjective ranges	Can provide predetermined ranges	Direct output of most simulations	Can be a direct output of algorithm

*Including range estimating and expected value methodologies

Figure 14: Principles of Estimation Techniques vs. Standard Techniques [1]

behind the use of an empirical estimating technique is that estimates should be as objective as possible and based on previous experience; this can be seen by the suggestion that methods which do not explicitly employ empirical estimates can be made to use empirical data through the use of lessons learned. [1] This requirement for objectivity exists because studies have shown the existence of significant optimism bias when individuals make conventional predictions about a future state of a project. [68, 67] This bias can be mitigated by taking an “outside” view of the project— asking and answering questions about a class of concept instead of a specific concept. [43] This leads to the principle which should be employed for novel concepts: a margin estimation technique should maximize objectivity and seek to take an “outside” view of the problem.

Another principle that must be included in this analysis is the flexibility of an estimation method to account for novel concepts. Novel concepts represent fundamentally new vehicles and concepts of operations, and a margin estimation technique needs to be able to account for this departure from the existing experience base.

A third principle for margin estimation is that a margin estimation technique should account for all of the relevant sources of uncertainty as identified in Chapter 2. If an estimation technique does not account for all sources of uncertainty, then the program risks potential mass growth or performance losses due to omitted uncertainties.

A final principle necessary for a novel concept is the same as an AACE principle— the need for providing probabilistic estimating results. Deterministic estimates and predictions provide an illusion of certainty in a decision maker’s mind. By comparison, probabilistic estimates communicate the relative uncertainty of a prediction to the decision maker. This output helps ensure that a decision maker is aware of the potential consequences of the decision at hand. As explained in Chapter 1, the decision as to the amount of design margin to carry forward in the development of a novel concept is critical to the success of a program; therefore, the information used in making this decision should be expressed probabilistically. [1, 72]

In the following sections, each family of margin estimation technique will be described in detail and evaluated with respect to these four principles as well as their historical accuracy.

3.2 Predetermined Percentage

3.2.1 Description

The simplest, and most common, method of determining a design margin is to add a predetermined percentage of margin to an existing performance or schedule estimate. [129, 128] In this method, the project team simply refers to a relevant standard or company policy and uses the appropriate design margin for a class of project. Numerous organizations such as the AIAA, NASA, and individual contractors have developed mass property standards. [6, 150] Because standards are put together as part of an effort to codify lessons learned, using a predetermined percentage is a simple way to make an estimate based on previous program experience. [21, 1]

The most basic application of this method imparts a single scalar value for design margin during requirements development or conceptual design. [1] More advanced standards go further. The AIAA standard, Figure 5, and the JPL standard, Figure 15, both specify a burndown rate of design margin throughout the phases of program development. [6, 93, 150] Further complications of this method involve specifying design margin estimations and depletion schedules for individual subsystems; this additional feature takes into account the fact that certain vehicle subsystems will experience additional mass growth and mature more slowly during the development process. [6, 102]

3.2.2 Evaluation

The standards published by numerous organizations are the result of studying historical projects and employing lessons learned. Through this development, this method satisfies the original AACE principle of employing empiricism. Furthermore, predetermined percentages give the design staff limited flexibility in the ultimate choice for a margin estimate. This limited choice ensures that this method takes an “outside” view and maximizes objectivity.

While the limited flexibility inherent in this method ensures objectivity, it severely limits the options of a design team in accounting for a novel concept. The use of historical data limits the use of this method to concepts within the dataset; its use on a novel concept would

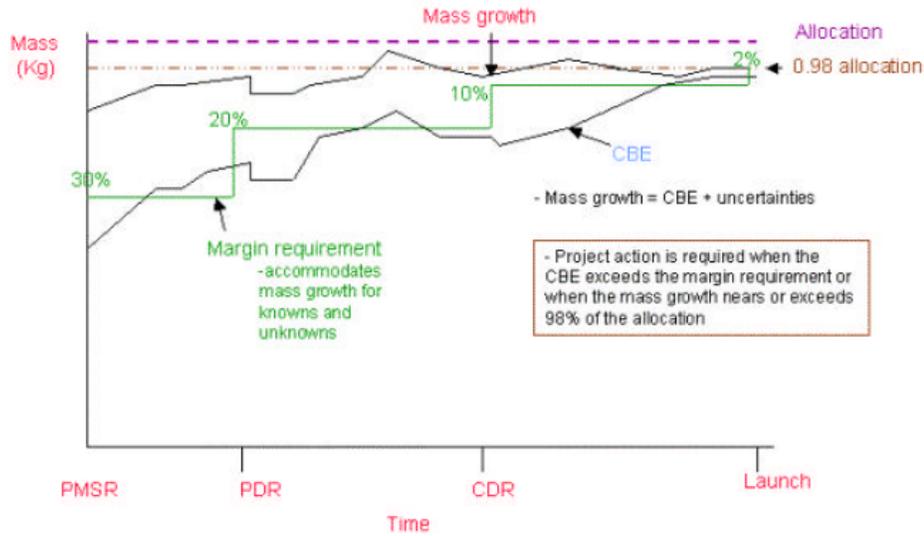


Figure 15: Jet Propulsion Laboratory Mass Growth Standard [150]

represent an extrapolation from existing experience. As Section 1.2.1 demonstrated, different classes of vehicle have different ranges of mass growth; applying an existing percentage from an existing class of vehicle could have a negative impact on a new vehicle.

Predetermined percentage methods do account for the all of the underlying sources of uncertainty. Because the guidelines are built up from lessons learned and historical programs, these programs experienced growth from all sources of uncertainty. Guidelines based on historical growth and lessons learned will therefore account for all sources of uncertainty as well.

Standards used to estimate margin either specify a specific percentage to be used, or specify a range of percentages. Standards which provide ranges meet the AACE principle for providing probabilistic results. While these ranges provide additional information to the decision maker, they fail to show the underlying sensitivities of the estimation uncertainty. Additionally, most standards or company policies only provide a single point estimate. These singular point estimates can imply an unjustified degree of certainty and, within the construction industry, have been described as “an educated guess at best.” [80]

A study conducted on the accuracy of the AIAA Mass Properties Control for Space Systems Standard as applied to the historical record of space development projects has shown

that 30% of historical program exceeded the recommended growth allowance of 32.5%. Furthermore, another 20% of historical projects incurred significantly smaller growth, 20% or less, than the AIAA recommended allocation. [130]

Due to the lack of flexibility in its ability to adapt to novel concepts, its limited capability to provide probabilistic estimations, and its historical inaccuracy, a predetermined percentage should not be used as a margin estimation technique for a novel concept. Efforts to address these shortcomings would either require more analysis or require the existence of a historical database of similar vehicles. Extending this method by adding more analysis would work against its strengths of being an easy-to-use, objective method for arriving at an estimation by adding additional analysis and decisions.

3.3 Historical Regression

3.3.1 Description

A more mathematically rigorous way of utilizing lessons learned from past projects is through the use of parametric historical regressions. Current design methods for most types of vehicles utilize historical regressions as a way to generate initial estimates. These parametric equations are known as mass estimating relationships (MERs); when used for cost estimation, parametric equations are known as cost estimating relationships (CERs). [34, 54, 74, 143, 107, 114] These relationships can take on a number of mathematical forms—linear or quadratic equations, neural networks, or logarithmic relationships. [4] The individual parameters within a relationship can be a function of almost any design parameter of interest as long as there is a statistically significant relationship. For example, Figure 16 presents the regression for spacecraft mass as a function of the number of man-days spent in space; the individual data points of historical NASA missions used in this regression can be seen on the chart.

The same data can be used during early design to estimate a design margin. Because the MERs are formed from a statistical regression of historical data, the model fit error of the historical points can be computed where error is defined by Equation 16. This error can be used to compute a correction factor for the mass estimate— a scalar which multiplies the

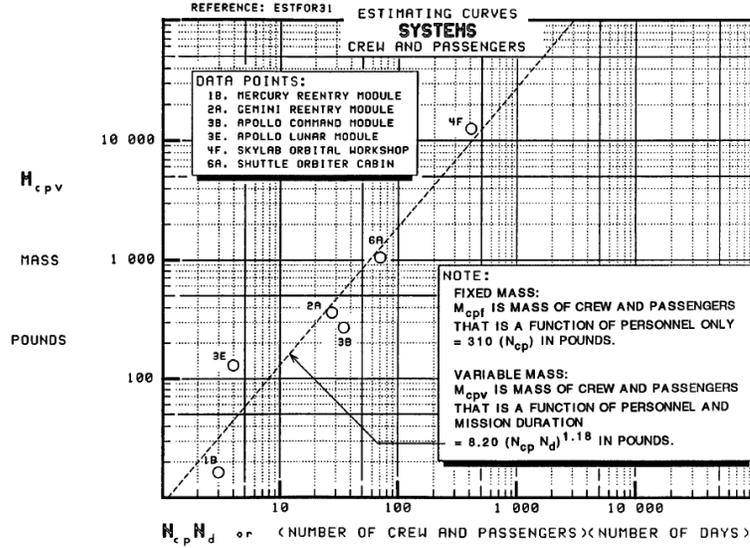


Figure 16: Spacecraft Mass versus Man Days in Space [54]

predicted mass in order to reconcile the actual and predicted mass values . This correction factor is defined in Equation 17. Because each historical point will have a correction factor, the sample mean and standard deviation of the correction factors for a given MER can be computed; Figure 17 shows a plot of a MER with +/- 1 standard deviation for the correction factor. With these statistics, a Monte Carlo simulation can be conducted which can produce a cumulative distribution function (CDF) of predicted mass. Given this CDF, a decision maker can select the necessary dry mass estimate based on relative risk tolerance. [24]

$$Error = \frac{M_{actual} - M_{predicted}}{M_{actual}} = \frac{Residual}{M_{actual}} \quad (16)$$

$$C = \frac{1}{1 - \frac{M_{actual} - M_{predicted}}{M_{actual}}} = \frac{1}{1 - Error} \quad (17)$$

Utilizing the statistical data from a MER regression is not the only way to regress historical data to estimate a design margin. A NASA study into design margins for the Constellation program collected historical estimates of mass at project SRR and launch. Based on this data, the study performed a regression where the dry mass at launch is a function of the SRR dry mass and the error statistics of the regression; the equation used

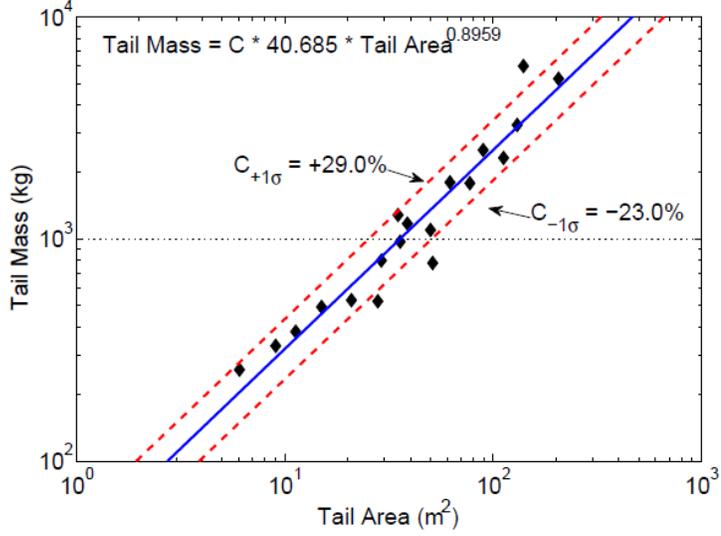


Figure 17: Vertical Tail Mass vs. Tail Area with Bounds of 1 Standard Deviation [24]

for this regression can be seen in Figure 18. Within this equation, β is the estimated mass growth coefficient and ϵM_{SRR}^{dry} is the sample error. Within this regression the sample error is scaled by M_{SRR}^{dry} in order to account for the fact that the historical database used for model creation contains data from both unmanned and manned missions. Because this regression is used in a probabilistic simulation, the distribution used to generate potential mass estimates can be seen in Equation 19. [89]

$$M_{launch}^{dry} = \beta M_{SRR}^{dry} + \epsilon M_{SRR}^{dry} \quad (18)$$

$$M_{launch}^{dry} \sim N((\hat{\beta} M_{SRR}^{dry}, \sqrt{\hat{s}^2(1+n^{-1})} M_{SRR}^{dry}) \sim N(1.29718 M_{SRR}^{dry}, 0.1579 M_{SRR}^{dry}) \quad (19)$$

Another study was conducted at the Air Force Institute of Technology in order to determine construction project cost margin. The study utilized data on 243 projects from a database of Air Force construction projects; of these projects 218 data points were used for regression and 25 were used for model validation. The resulting regression produced an equation for the anticipated construction cost margin as a function of ten parameters. These ten parameters described the project's overall characteristics, its design, and aspects

about the contracting mechanism. [128]

A method known as reference class forecasting, while not utilizing a parametric regression, falls under this category as it seeks to establish a probability distribution based on historical data. This method has three steps. First a relevant class of past projects must be identified; this list must be statistically significant but sufficiently narrow to enable comparison. The next step is to establish a probability distribution based on the selected past projects. This distribution tracks the overall cost overrun of these projects. The third step in this process is to compare the specific project to the relevant distribution to determine the necessary margin. [43]

3.3.2 Evaluation

Historical regressions are a more sophisticated way of utilizing historical data in order to make projections about new development projects. Because historical data is explicitly used as the sole input, this family of methods employs empiricism satisfying the original AACE principle.

Historical regressions also account for all of the underlying sources of uncertainty. Like the methods of predetermined percentage, this family relies on historical data. The individual projects from which parametric models are built incurred all of the sources of uncertainty; therefore, any prediction made from these parametric models will also account for the same sources of uncertainty.

Unlike methods of predetermined percentage, historical regressions enable the use of probabilistics by decision makers. Because parametric models are statistical regression models constructed from historical data points, the underlying goodness of fit statistics are available for these models. For a given prediction, the confidence intervals corresponding to a desired level of certainty can be calculated. Utilizing this capability, a decision maker can examine the relative impacts of varying confidence levels to make a more informed decision. [4]

Estimations based on historical data have shown good predictive capability. The Air Force Institute of Technology study's model matched the validation data points very closely.

[128] Another study found that estimates based on historical estimates had slightly better predictive capability than the other three families of margin estimation. [21]

The biggest shortcoming of this method is that it relies explicitly on historical data. In order to make a prediction about a vehicle, previous vehicles of a similar type must have been previously developed. This requirement breaks down in the case of a novel concept as there is no historical database of similar concepts. Therefore, this method is not only not flexible enough to handle novel concepts, but also the lack of historical points makes the application of this method impossible.

3.4 *Expert Opinion*

3.4.1 Description

A more flexible alternative to utilizing predetermined percentages or historical regression is utilizing expert opinion. In this method an organization's subject matter experts are asked to use their experience and judgment to determine an appropriate margin estimate. This estimate can be based on an analysis of a particular project's specific risk drivers as well as the completeness of the original estimate. [21, 1]

Expert opinion is also highly leveraged in hybrid methods— most methods used in industry employ some form of expert opinion. Expert opinion can be used in combination with predetermined percentages where experts can assign risk levels associated with a predetermined contingency allocation. Expert opinion also plays a large part in more structured organizations where experts provide the estimates for values which can be used as part of a probabilistic technique. [21, 1]

3.4.2 Evaluation

The biggest strength of expert-based methods is its flexibility. The experts employed in making these judgments can make recommendations based on the specifics of a novel concept. Free from a reliance on historical data, this family of methods will take into account the new technologies, designs, and concepts of operations inherent in a novel concept development program. The utilization of project-specific factors can lead to more accurate estimates when compared with methods which ignore project specifics. [21]

Expert opinion methods can also take into account the relevant sources of uncertainty. A team of experts will have past experiences with previous projects including projects which were novel concepts at the time of their development. This experience should provide the team with a background on the relevant sources of uncertainty that occurred in previous projects and are likely to occur for a new novel concept development.

Most expert-based methods do not provide probabilistic estimations but rather focus on determining a deterministic estimate of margin. Like methods of predetermined percentages which provide deterministic values, a deterministic forecast communicates a level of certainty to the forecast which can misrepresent the underlying uncertainties inherent in a forecast. Alternatively experts can provide ranges to be fed into a simulation analysis, but these methods are classified as probabilistic techniques and will be analyzed in the following section.

The ability of an expert-based method to address project-specific considerations lies in its subjectivity— this subjectivity is also the main disadvantage in using expert opinion. [21] This subjectivity has typically led to inaccurate forecasts through two mechanisms— strategic misrepresentation and optimism bias. Within the world of cost and schedule estimation, pressure is placed on projects during early design to show favorable numbers in order to win approval in the face of competition for resources. This leads to the intentional under-estimation of cost and schedule metrics. [43] Decision makers also fall victim to the planning fallacy where decisions are based on overestimating benefits and underestimating costs. This optimism stems from cognitive biases which affect how individuals process information. [78]

The subjective nature of this method directly leads to another weakness: the reliance on experts within an organization. For any problem, the number of experts qualified to make a judgment will be limited. Additionally, the transfer of expertise from experts to new team members or employees is very difficult. [21] An organization which decides to maintain the capability of utilizing expert opinion will therefore need to retain the skills of its subject matter experts even in the absence of new development projects. When taking into account the need to estimate a margin for a novel concept, the problem only intensifies

as there will likely be even fewer experts qualified to make a judgment of a novel concept. This will likely lead to the situation where the same people who made the initial projections of performance will also be estimating the degree to which those initial projections were incorrect.

Within the world of cost estimation, margin estimation through expert opinion has proven to have good predictive capability. [21] However, it is worth noting that this study involved estimations of relatively routine projects, not novel concepts. Because of its use on routine projects, the weaknesses due to the subjectivity of expert opinions will be mitigated—more experts with better knowledge of the project will be available. Furthermore, optimism bias, while still existent, is tempered because of the lack of new technologies and new concepts.

Overall, the human element is at the center of both the strengths and weaknesses of expert-driven methods of estimating design margin. Greater flexibility and applicability to novel concepts is gained at the expense of probabilistic analysis and objectivity. Unlike predetermined percentage and historical regression methods, no show-stoppers exist in implementing an expert-driven method for use with a novel concept.

3.5 Probabilistic Techniques

3.5.1 Description

As a family of methodologies, probabilistic techniques enable the quantitative estimation of risk and can subsequently provide a risk-informed design margin estimation. These techniques commonly employ Monte Carlo simulation to generate a statistical distribution of outcomes. Based on this distribution, a decision maker can determine an appropriate level of risk and corresponding design margin. [21]

This family of techniques includes the expected value method, range estimating, and range estimating applied to analysis tools. Additionally, the application of these methods in aerospace studies will be analyzed.

3.5.1.1 *Expected Value Method*

The simplest probabilistic technique used to determine design margin is the expected value method. In this method, individual risk drivers and corresponding risk events are identified. For each risk event, the probability of that risk occurring as well as its impact must also be identified. The expected value of the impact of that event is expressed in Equation 20. [3]

$$\text{Expected Value} = \text{Probability of Risk Occurring} * \text{Impact if It Occurs} \quad (20)$$

This method can be used to provide a risk estimation without the use of a Monte Carlo simulation. However, the AACE standard recommends the use of a Monte Carlo simulation to more fully explore the implications of each risk event. When used with a Monte Carlo, the probability of occurrence as well as the impact can be specified using probability distributions. Additionally, dependencies between different risk events can be specified so that events do not have to be independent. [3]

The AACE standard for the contingency determination using the expected value method warns that this method should not be used alone. The expected value method is very appropriate for project-specific risks which can be identified and estimated by the project team. Estimates for systematic risks (risks due to project complexity, technology, culture, etc.) should be calculated using a different estimation methodology. [3]

3.5.1.2 *Range Estimating*

A more sophisticated family of probabilistic methods for margin estimation is range estimating. Within this family of methodologies, the vehicle or project under consideration is broken down into a work breakdown structure. A work breakdown structure subdivides a vehicle into constituent subsystems or a project into subtasks which must be completed. Based on this breakdown, the potential weight, cost, or duration of each item can be estimated— this estimate usually takes the form of a triangular distribution where the estimator can proscribe a minimum, maximum, and most likely value. [2, 26] In the simplest of these methods, the subitems of the work breakdown structure are considered independent. [9, 26] Given a distribution for each subitem of the work breakdown structure, a Monte

Carlo simulation can be conducted to define the overall distribution of the project or vehicle. Based on this output, a nominal design weight or cost as well as a margin can be determined. [9, 19, 34, 46, 123, 139, 151]

This method has been subsequently refined in the cost engineering community. The first key refinement redefines the estimation phase. The current AACE recommended practice for contingency estimation through range estimation states that *only* critical items within the WBS should be ranged; critical items are defined as “one whose actual value can vary from its target, either favorably or unfavorably, by such a magnitude that the bottom line cost (or profit) of the project would change by an amount greater than its critical variance.” [2] This recommendation implies an additional step within this family of margin estimating techniques—performing a sensitivity analysis and identifying critical items. Furthermore, the recommended practice cautions that if non-critical items are ranged, the resulting estimate will exhibit a narrower predicted range of potential costs and can result in an understatement of required contingency. [2, 28]

The process of estimating ranges for individual items of the work breakdown structure has also received refinements. The current recommended practice for determining the three point estimate recommends that the optimistic and pessimistic estimates should only account for likely extremes. Highly unlikely outcomes should be omitted from the range as they can greatly skew results. The recommendation is that the extremes of the range should represent the 1% and 99% probability limits. In other words, if an estimate has less than a 1% chance of occurring, then that estimate should be omitted from an individual item’s range. [2, 28]

In addition to recommendations on the ranges for individual items, additional work has been done in investigating the probability distributions used for representing ranges. While the triangular distribution is the most popular distribution, other probability distributions can change the behavior of the overall risk analysis. If a team wants finer control over the skewness of the distribution, a double triangular distribution can be used. A double triangular distribution, Figure 18, is comprised of two triangular distributions— one distribution representing the probability of an overrun and another triangular distribution representing

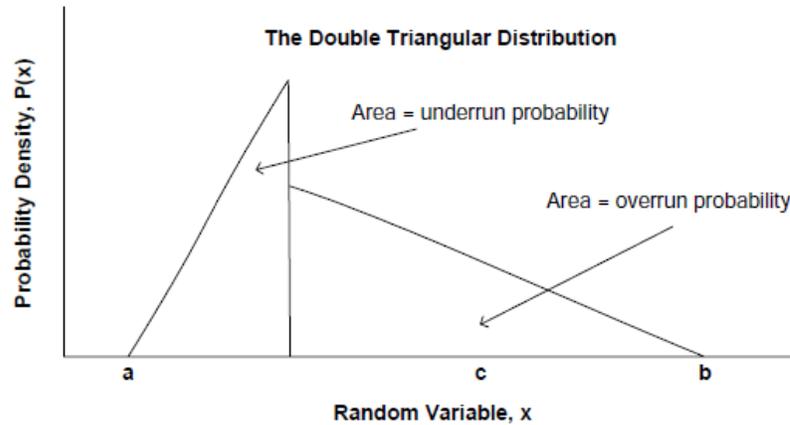


Figure 18: Double Triangular Distribution [2]

the probability of an underrun. The AACE recommended practice for range estimating recommends the use of the double triangular distribution in most cases. [2] In addition to the double triangular distribution, the PERT distribution has seen use in range estimating. The PERT distribution is a modified Beta distribution which is a function of three parameters: the minimum, the most likely, and maximum values. These parameters are the same parameters which define a triangular distribution which allows for easy integration into existing methods as only the underlying distribution needs to be changed. A PERT distribution's standard deviation is more sensitive to the most likely value than to the minimum and maximum values. By comparison, a triangular distribution's standard deviation is equally sensitive to the most likely, maximum, and minimum values. Because of this, PERT distributions have a systematically smaller standard deviation than triangular distributions especially when the input parameters produce highly skewed distributions. The use of a PERT distribution in an additive model will result in 10% less uncertainty than an equivalent model using triangular distributions. [139]

A key assumption in most implementations of range estimating is the assumption that each line of a work breakdown structure can be simulated independently of one other. This assumption greatly reduces the effort required for conducting a risk analysis because the analysis team does not have to define correlations. However, examinations of this assumption have shown that the modeling of correlation is necessary to improve the accuracy

of the resulting Monte Carlo simulation because it will likely underestimate the true variance of the project being analyzed. [60, 26] The most basic way of implementing correlation into a Monte Carlo simulation is through the use of correlation matrices. A correlation matrix is a symmetric matrix which maps each item in a Monte Carlo simulation to one another; the individual elements of this matrix are rank order correlation coefficients. These coefficients can have a range from -1 to +1 where -1 represents an exact negative correlation, 0 represents independence, and +1 represents an exact positive correlation. [139]

3.5.1.3 Range Estimating Applied to Schedule Estimation

A similar, but more complicated, methodology can be applied to scheduling. A schedule can be broken down into its constituent tasks, and the methodology of range estimating can be applied to the individual tasks. Additionally, the dependencies between tasks must be explicitly defined. The following dependencies between tasks have been identified: [139]

1. A new task cannot start until another task has completed (finish–start)
2. A new task cannot start until another task has started (start–start)
3. A new task cannot start until another task has been partially completed (start–start + lag)
4. An existing task cannot finish until another has finished (finish–finish)
5. An existing task cannot finish until another has reached a certain point in its execution (finish–finish - lag)

After the problem is defined and a Monte Carlo simulation is performed, a likely critical path can be identified. The critical path is the sequence of tasks within a schedule which define the maximum time required to complete a project; in contrast, subcritical paths contain activities which can be performed in parallel to the critical path and should not affect the total time required for completion. A margin is only applied to tasks on the critical path, and the sum total of the individual task margins is equal to the margin for the entire project. [8]

3.5.1.4 Range Estimating Applied to Models

Another methodology for margin estimation is formalized by Thunnissen as part of his PhD thesis. This method begins by identifying tradable parameters—these are parameters which are important in satisfying the system’s requirements and can be expended to make up for a deficiencies within this set. The next step in the methodology is to create analysis models capable of analyzing the necessary tradable parameters. These models will have numerous inputs whose uncertainty needs to be quantified. These steps differ from other range estimating techniques because this method involves using probability distributions as an input to an analysis model. Once this uncertainty is quantified, a Monte Carlo simulation (or similar stochastic method) is performed on the analysis model. This output can be further analyzed to determine an acceptable level of risk and a corresponding margin. [131, 132]

3.5.1.5 Use in Aerospace Applications

While probabilistic methodologies have seen more use in other disciplines, some studies have been conducted on the probabilistic estimation of margin in space and launch vehicles. Unlike cost or schedule margin estimates, the design margin of a space or launch vehicle feeds back into the sizing of the vehicle—the additional weight triggers a scaling up of the vehicle. This larger vehicle will need to be reclosed by adding additional propellant and corresponding tanking in order complete the required mission. [36]

In order to examine the effect of uncertainty on a lunar lander, engineers at SpaceWorks Engineering Inc. performed a study which applied the principles of range estimating. The sizing tool, Moonraker, allows for input parameters to modify the weight assumptions. The study utilized a level 1 weight breakdown structure, only delineating the top level of subsystems. For each subsystem, a triangular distribution of percentage weight growth was identified, and for all distributions, the likeliest value was set at 0% growth. A Monte Carlo simulation was then used to quantify the uncertainty, and a design weight corresponding to a 90% confidence level was selected. This corresponded to a 8.22% and 13.15% growth in the dry mass of the ascent and descent stages respectively. [140]

A study examining the effects of uncertainty on single stage to orbit and bimese launch vehicles was performed by Wilhite et. al. This study utilized the commercially available tool @Risk and the Launch Vehicle Sizer and Synthesis tool. MERs were used to generate an initial estimate of component weights, and uncertainty was applied to these estimations as well as engine I_{sp} , aerodynamic drag, and packaging efficiency. This study utilized uniform distributions of uncertainty. Utilizing these distributions, @Risk performed a Monte Carlo simulation through the Launch Vehicle Sizer and Synthesis tool. At a confidence level of 95%, this study showed growth of 25% and 31% for the bimese and single stage concepts respectively. [146]

In support of the Constellation program, a study was conducted to determine the performance requirements of the Ares V launch vehicle. Because the requirements of the Ares V depended on the entire architecture of the Constellation program, three separate uncertainties had to be accounted for: the uncertainty of the dry mass of Constellation's in-space elements, the performance of the in-space propulsion, and the performance of the Ares V. The uncertainty of the dry mass was accounted for using historical regression as described in Section 3.3.1. The uncertainty bounds for in-space propulsion elements were modeled using a triangular distribution for I_{sp} . The uncertainty in Ares V performance was modeled using a beta distribution; the parameters of this beta distribution were derived by expert opinion which was informed by model runs of a synthesis and sizing environment. These three estimations of uncertainty were combined in a single Monte Carlo simulation to determine the total payload that could be landed on the lunar surface for a reference mission. Unlike the SpaceWorks and Wilhite studies, this study focused on discrete vehicle concepts as opposed to rubberized vehicles. This study concluded that in order to have sufficient margin for lunar missions, the Ares V should use 6 RS-68 engines and a 5 or 5.5 segment solid rocket booster instead of the original design which featured 5 RS-68 engines and a 5 segment solid rocket booster.. [89]

3.5.2 Evaluation

Probabilistic methods represent a very flexible way to analyze concepts. Common steps in probabilistic methods are the selection of a work breakdown structure and subsequent uncertainty estimation. These steps allow a risk analysis team to customize a study to a particular vehicle. This flexibility allows a team to accommodate novel concepts, new technologies, and different concepts of operation.

By its nature, probabilistic estimating techniques will produce probabilistic results. These results will allow decision makers to see the sensitivities of the system to uncertainty. Furthermore, decision makers can select a specific level of confidence in order to determine the appropriate design margin.

Most probabilistic methods involve the elicitation of expert opinion in order to generate input probability distributions. Because individuals are involved, the question of subjectivity and bias arises. The step of breaking down a system into its subsystems and components mitigates this bias as an expert is forced to think of smaller pieces of the overall system. This thinking of smaller pieces of the puzzle will also improve the accuracy of the estimate. When studying subjective cost estimates, it was found that “in general the reliability of the subjective estimates decreases as the subsystem increases in size and complexity since the cost estimator is less able to deal with complicated systems than with simple systems.” [26]

A study into the accuracy of probabilistic techniques as applied to cost estimation has shown that the accuracy of this family of estimating techniques is highly dependent on the level of project definition. For well-defined projects, probabilistic techniques outperform expert judgment and predetermined percentages. However, for poorly defined projects, probabilistic techniques perform poorly—both the median difference of predicted vs. actual and the corresponding variance are much larger than any other form of estimation. When the three other families of margin estimating techniques were analyzed by breaking out well and poorly defined projects, none of the other methods showed a dramatic decrease in performance. This decrease in performance appears to be unique to probabilistic methods. [21]

At first glance it is unclear why probabilistic methods under-perform other margin estimation techniques for poorly-defined projects. This is especially interesting when one considers that the same experts who provide insight for estimations based on expert opinion are likely the same experts who would provide input distributions for a probabilistic method. However, the answer to this discrepancy is apparent when considering the underlying sources of uncertainty as outlined in Section 2.3. The act of splitting a project into its subsystems and components implies a baseline vehicle or project and focuses the attention of an estimator on a small portion of that baseline. This breakdown fundamentally changes the nature of the uncertainties under consideration; the focus on individual components and subsystems only allows an estimator to consider model uncertainty and some technological uncertainty. The implicit baseline in the breakdown does not allow estimates to be made on human errors, changes in technology as well as any source of exogenous, phenomenological, or volitional uncertainty—uncertainties which would fundamentally change the baseline design cannot be considered if the system is decomposed into its constituent subsystems and components.

The observation that range estimation does not account for potential baseline changes has been noted in the literature. In a study comparing the results of range estimating to a naval development program, Boze found that the “method will not sufficiently forecast weight values for unanticipated vehicle physical configuration changes ... or for changes made to performance specifications.” [18]

Overall, the greatest strengths of this family of estimating methods are their ability to specify probabilistic results and their flexibility. Furthermore, these methods provide a way to mitigate the subjectivity of the expert opinions. These advantages are gained at the expense of being able to account for uncertainties which change the baseline implicit in the estimations. Current methodologies are more suited for programs which have a more mature baseline.

3.6 Gaps

Of the four families of margin estimating techniques, predetermined percentage and historical regression cannot be applied to a novel concept. For historical regression, as no previous examples of a new vehicle concept exist, a regression cannot be performed. This lack of historical data acts as a show-stopper for the utilization of this methodology. In the case of predetermined percentages, this method is also reliant on historical data; furthermore, it is incapable of providing probabilistic results or being flexible enough to capture new ideas inherent in a novel concept. Finally, Thompson showed that predetermined percentages are not accurate. [130]

In addition to these methods not being suited to novel concepts, not much can be done to extend these methods to make them more applicable. Regression analyses will always depend on the underlying data; in the absence of data applicable to a new class of vehicle, any extensions or changes to this family of methods will not address the problem. Predetermined percentages could be applied to a novel concept if one could answer the question, “what is an appropriate margin estimate for a new class of vehicle?” As with historical regression, the lack of historical experience makes answering this question impossible.

Expert-based methods showed no show-stoppers; this class of methods can be applied to novel concepts. Similarly, the analysis of probabilistic methods has shown that this class of methods can also be applied to novel concepts. However, each family of methods has unique shortfalls. Expert-based methods cannot provide probabilistic estimates and are subject to the subjective opinions of experts. Most importantly, the weaknesses of expert-based methods lie in the ability of a company to develop and retain a core group of employees qualified to make these judgments. Probabilistic methods employ experts while mitigating the subjectivity of their opinion by breaking the problem down into individual subsystems and components. However, this breakdown works to obscure sources of uncertainty which would have otherwise been addressed by experts.

The drawbacks in expert-based methods and probabilistic methods represent gaps in current analysis capabilities. These gaps must be addressed so that a margin estimation technique for a novel concept can embody the principles described in Section 3.1.2.

The first identified shortcoming of expert-driven methods is a lack of probabilistic estimation capability. Simply asking experts to provide a subjective probability distribution is not a useful extension because studies have shown that individuals tend to overestimate the effect of infrequent events. [42] This can be mitigated by breaking the overall estimation problem into smaller, easier to estimate subproblems, but this is problematic because what was an expert-based method is now, by definition, a probabilistic method.

If an organization decided that probabilistic estimation capability was not a necessity and this shortcoming of expert-driven methods could be ignored, this family of methods still has shortcomings due to the nature of employing experts. The largest problem has to do with the limited number of experts available within a company to analyze a novel concept. Due to the lack of experience with untried concepts, the experts available will likely be the same people that designed the current vehicle. These experts are likely to have an “inside view” of the problem as they will have invested time and effort as part of the development team.

These two shortcomings of expert-driven are driven by the fact that, ultimately, individuals are responsible for the method’s success or failure. While an extension could possibly be made to extract probabilistic outputs from experts, no technical fix can be applied to address a lack of experts knowledgeable of a novel concept. Given that this is a human resources problem, probabilistic techniques will be examined to determine if an extension can address its shortcomings.

The major shortcoming of probabilistic methods is its inability to capture all of the relevant forms of uncertainty. This shortcoming is summarized in Observation 4.

Observation 4: Probabilistic techniques show promise in their ability to account for novel concepts, but current techniques do not capture all relevant forms of uncertainty. Current techniques rely on a bottom-up approach which cannot address phenomenological, volitional, exogenous uncertainties, and discrete technological uncertainties.

The shortcoming noted in Observation 4 shows that the existing methods do not completely cover relevant sources of uncertainty. Unlike expert-driven methods, the underlying shortcomings of this family of methods are technical. Addressing these shortcomings is the

subject of Research Question 1.

Research Question 1: How can existing probabilistic margin estimation techniques be augmented to account for sources of uncertainty which are not adequately addressed by a bottom-up formulation?

This research question clarifies the research objective of this thesis. The examination of existing ways to estimate margin has shown that probabilistic formulations offer the best starting point for the creation of a methodology to evaluate novel concepts. If the shortcomings of probabilistic methods can be addressed by an methodological extension, then the research objective of creating a methodology for estimating the uncertainty of a novel concept will be achieved.

The proposed extension for probabilistic analysis will be discussed in Chapter 4.

CHAPTER IV

EXTENDING PROBABILISTIC ESTIMATION TECHNIQUES

As was described in Section 3.6 and Research Question 1, the family of probabilistic estimating techniques shows promise in its ability to forecast uncertainties with novel concepts. The shortcoming of this family of methods is that its current formulation does not allow for it to account for all of the sources of uncertainty that affect space and launch vehicles as outlined in Section 2.3. This chapter will explore possible extensions to probabilistic estimating techniques with the potential to address additional forms of uncertainty.

4.1 Application of Range Estimating to Potential Baseline Changes

The most basic extension of range estimating is to apply the method to different potential baselines. If the configurations to be ranged cover the breadth of potential production systems, then the forms of uncertainty as described in Section 2.3 will be addressed by this simple extension.

The key aspect of this extension is the ability to range a large number of configuration changes; as the capability to examine different configurations decreases, the likelihood that significant changes will be omitted increases. This aspect of a simple extended method leads to the following hypothesis.

Hypothesis: If evaluating potential alternative baselines through range estimation is a viable method, then the relevant sources of uncertainty affecting space and launch vehicles will be addressed by probabilistic methods.

In order to test this hypothesis, a combinatorial study of the historical changes to the Space Shuttle Orbiter was conducted to determine the level of effort necessary to apply range estimating to alternative configurations.

4.1.1 Space Shuttle Combinatorial Study

As Section 1.2.4 describes, standards for margin estimation call for a margin to be assigned at ATP. In order to account for potential baseline changes through range estimation, each

potential change to the baseline configuration will need to be examined. Furthermore, as range estimating requires individual subsystems to be evaluated, for each unique combination of the morphological matrix, the individual subsystems of the Orbiter which need to be reexamined will need to be determined.

In order to determine the total number of reexaminations necessary to account for the Orbiter's historical changes, two levels of analysis are necessary. First, the major historical changes to the Orbiter must be documented and compiled into a matrix of alternatives to determine how many possible individual Orbiters can be produced given the potential design changes. Second, the internal subsystems of the Orbiter must be identified and linked to the Orbiter's historical design changes.

4.1.1.1 Changes to Shuttle Baseline design

The Space Shuttle Orbiter experienced a significant number of design changes between ATP and production. The original Orbiter at ATP included a 3,220 square foot blended delta wing, payload bay mounted air breathing engines, 2 abort solid rocket motors (ASRM), 4 hydraulic systems, an exposed forward orbital maneuvering system (OMS), a centrally mounted aft OMS, and a total length of 125.8 feet. The first changes made to the Orbiter were the removal of the ASRMs and the air breathing engines; additionally, the aft OMS pod was moved to the shoulder of the aft fuselage and overlapped the payload bay. Another large redesign of the Orbiter came next as the 150K orbiter was redesigned to have a dry weight (with growth) of 150,000 lbs. This design came with a new, Lockheed-inspired, double-delta wing of 2,690 square feet and a slightly shorter fuselage of 125.2 feet. Additionally, the air-breathing propulsion system reappeared as wing-mounted, removable engines to support ferry missions. The previously-deleted ASRM system was restored to the baseline. After this re-design, further changes were made including a reduction in the overall length of the Orbiter to 122.8 feet accompanied by changes to the wing planform. Final changes in 1974 included a final removal of the ASRMs and the deletion of air-breathing engine accommodations. Furthermore, a parachute breaking system was added, the aft OMS pod was redesigned to accommodate a new payload bay door, the forward OMS system was

Air-Breathing Engines	none	Payload Bay Mounted	Wing Mounted	
ASRM	none	two		
AFT OMS Pod	Center-Fuselage	Fuselage Shoulder/overlapping	Fuselage Shoulder	
Forward OMS	Shrouded	Exposed		
Wing Planform	Blended Delta	150 K double delta	V4/5 Double-Delta	
Fuselage Length	122.2 ft	122.8 ft	125.2 ft	125.8 ft
Hydraulic Systems		3	4	
Parachute	none	Single		

ATP Baseline System

Figure 19: Morphological Matrix of Historical Changes to the Orbiter

unshrouded, and the overall fuselage length was reduced to 122.2 feet. [96, 61]

In order to determine the total number of potential designs based on actual historical design changes, each change was entered into a morphological matrix. A morphological matrix is a matrix comprised of rows and individual entries. Each row represents a different category of alternatives, and individual entries belong to a category of alternatives. Selecting a single entry in each row defines a unique combination of alternatives. The total number of alternatives is a multiplicative combination of the number of alternatives in each row. [111, 110] The morphological matrix containing potential Orbiter designs can be seen in Figure 19; there are 1,728 unique configurations derived from this matrix. However, this number only identifies the total number of configurations which need to be evaluated; in order to determine the total number of range estimating reevaluations which need to take place, individual subsystems need to be considered.

4.1.1.2 Structure of the Orbiter

The Space Shuttle Orbiter has a very large weight breakdown structure covering numerous subsystems: structures, power, avionics, environmental control/life support, propulsion, and aerodynamic controls. In order to simplify the analysis, only the structural subsystems will be included in this analysis. The structural subsystem is most relevant because structures are more likely to be resized in the event of a baseline change. Furthermore, if reevaluations due to changes in the structural subsystem are too numerous, then the additional reevaluations of other subsystems will contribute to the result that range estimating

Table 9: Linkages Between Major Changes and Substructures

Design Change	Substructure Affected
Air-Breathing Engines	Mid Fuselage* or Wing**
Aft OMS Pod	Aft Fuselage
Forward OMS Pod	Forward Fuselage
Wing Planform	Wing
Fuselage Length	Forward Fuselage
Hydraulic Systems	Aft Fuselage
Parachute	Aft Fuselage

*Payload bay mounted engines

** Wing mounted engines

applied to numerous baseline changes is not viable.

The structural subsystem of the orbiter is primarily divided into three segments: the forward fuselage, the mid fuselage, and the aft fuselage. The forward fuselage is divided into upper and lower portion; the upper portion contains the crew compartment while the lower portion contains the forward OMS pod and the nose gear. The mid fuselage contains the structures which support the payload bay, payload bay doors, and main landing gear; the mid fuselage also interfaces with the forward fuselage, aft fuselage, and the forward section of the wing box. The structure of the wing is supported by the wing box which in turn interfaces with both the mid and aft fuselage. Finally, the aft fuselage contains the thrust structure and the aft OMS pods; this portion of the fuselage interfaces with the mid fuselage and the wing box.[70]

The linkages between the historical design changes described in Section 4.1.1.1 and the individual substructures can be seen in Table 9. While these linkages will translate the design changes to individual substructures, additional reevaluations will be necessary as the design change cascades throughout the vehicle. The total number of reevaluations necessary will be a function of the design load cases and the connectedness of the Orbiter's substructures.

In this study, the assumption will be made that the Orbiter experiences three major sizing cases— launch, gliding, and landing. During launch, the weight and acceleration of the Orbiter will be transferred to the thrust structure. During the gliding segment of flight,

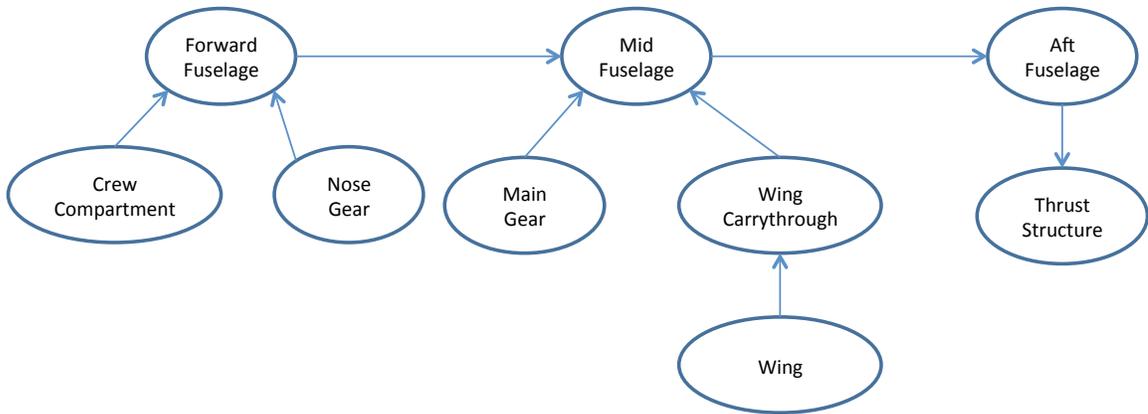


Figure 20: Space Shuttle Launch Loads DAG

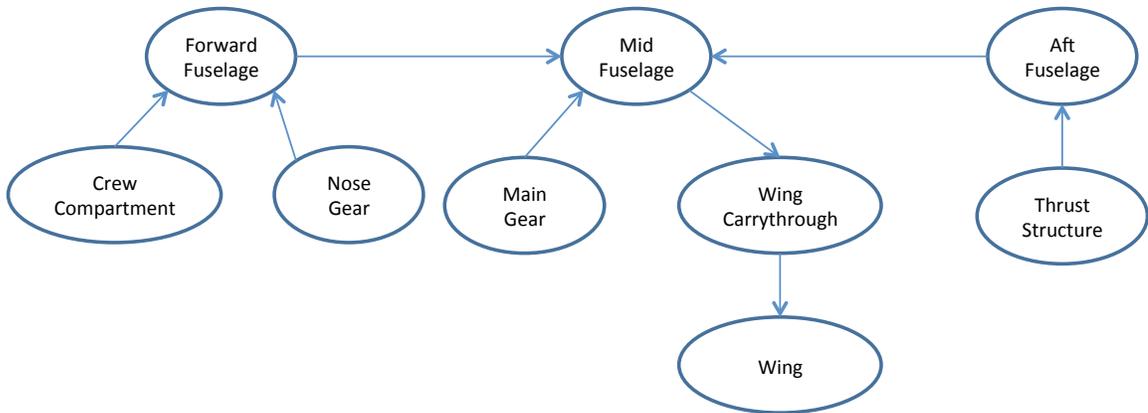


Figure 21: Space Shuttle Gliding Loads DAG

the Orbiter's weight will be transferred to the wing. Finally, during landing, the landing loads will be transferred to the landing gear attach points.

Based on these three simple load conditions and the description of the shuttle's substructure, three directed acyclic graphs (DAGs) were created to determine the cascading effect of a change in a single substructure. Figures 20, 21, and 22 show the DAGs for launch, gliding, and landing respectively. If a single substructure is tagged as needing a reevaluation, then the additional substructures downstream of the originally triggered structure also require reevaluation. For an example launch load case, if the wing requires a reevaluation, then the wing box, mid fuselage, aft fuselage, and thrust structure also require reevaluations.

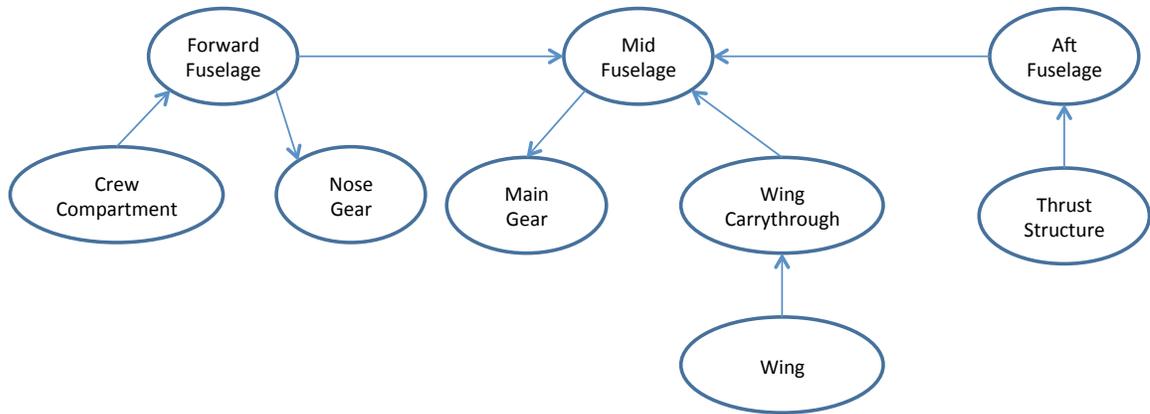


Figure 22: Space Shuttle Landing Loads DAG

4.1.1.3 Study Results

In order to determine the total number of reevaluations necessary, the morphological matrix was stepped through so that every combination of potential alternatives was selected. Each combination of alternatives represents a unique design. If a design contains an alternative not in the original baseline, then that alternative’s mapping to the substructure is triggered; since it is possible for a design to contain multiple selections not in the baseline, the linkage will be triggered for each alternative not in the original baseline. Each time a linkage is triggered, the DAGs for each load case will be triggered to determine the number of subsystems which will need to be reevaluated. If a subsystem is triggered during any of the three load cases, then it is flagged as needing reevaluation.

This analysis led to the result that 11,868 subsystem reevaluations need to take place in order to apply range estimating to the alternative baseline concepts identified by historical shuttle design changes. Assuming that an engineer would spend 15 minutes conducting a simple analysis to provide a range on a subsystem, the total required effort to apply this method is 2,967 man-hours. This represents a significant investment in time and effort on behalf of the project office. Furthermore, 2,967 man hours represents an optimistic lower bound on the amount of time necessary to complete an analysis because the underlying assumptions of this combinatorial study have significantly reduced the dimensionality of the problem. The effort required will likely be far greater as additional configurations, load cases, and subsystems are considered. Because of the required effort, the hypothesis that

range estimating can be applied to different baseline configurations is rejected, and a new method for addressing baseline changes will have to be adopted.

4.2 A Hybrid Methodology

The previous section showed that the sole reliance on range estimating is a non-viable way to address potential baseline configuration changes. However, the literature search conducted in Chapter 3 found that the other three methods of estimating design margin accounted for uncertainties related to baseline configuration changes. The fact that every other method of estimating margin does not have this shortcoming begs the question, “which characteristics of these three families of methods make them more able to account for all relevant uncertainties?”

The answer to this question can be seen in the first principles of these methods. Pre-determined percentage, historical regression, and expert opinion all take a holistic view of the program when estimating margin. This top-down view of the problem is what enables the underlying uncertainties to be characterized. In comparison to these three methods, range estimating takes a bottom-up approach to estimating margin by breaking the problem down into its component subsystems and components. As was found in Section 3.5.2, this bottom-up approach leads to the inability to consider all of the relevant sources of uncertainty.

The top-down view of the problem shared by other methods enables the consideration of all forms of uncertainty. In order for a probabilistic methodology to account for baseline changes, the bottom-up range estimating techniques need to be combined with a top-down technique that can estimate the effect of baseline changes. This will result in a hybrid methodology which combines the ability of range estimating to provide a high-quality probabilistic estimate with the ability to account for the relevant forms of uncertainty. This idea is captured in the following hypothesis:

Hypothesis 1: If a hybrid process is adopted with a bottom-up and a top-down component, then the relevant sources of uncertainty affecting space and launch vehicles will be quantified to enable a more complete estimate than can be constructed via existing methods of design margin estimation.

This hypothesis serves as the basis for a proposed methodology which can extend traditional probabilistic techniques to account for all relevant sources of uncertainty. This hypothesis will be accepted if a sample problem can be used to demonstrate the viability of this method. This hypothesized method, along with additional hypotheses which help further define the method, will be developed throughout the remainder of this chapter.

4.2.1 Methodology Development

A hybrid process recognizes that an overall margin allocation requires two different types of uncertainty analysis: traditional weight engineering which seeks to predict the eventual weight of a baseline and a risk/opportunity analysis which seeks to determine potential changes to the program. Noting this distinction, a bottom-up process is necessary for answering the traditional weight engineering problem while a top-down process focused on forecasting the uncertainty due to vehicle baseline changes is performed in parallel. Both methods will create estimates which will ultimately be combined to determine the overall design margin for a vehicle. An analogy can be made to the AIAA's Mass Properties Control for Space Systems Standard where separate estimates are recommended for in-scope and out-of-scope growth. [6] This hypothesized methodology will use a bottom-up method to account for in-scope sources of growth due to model and certain technological uncertainties. A top-down process will attempt to account for all other sources of uncertainty.

Range estimating has been selected to account for model and certain technological uncertainties within a hybrid methodology because it is an established method that can extract high quality estimates of a particular vehicle. The estimation produced through range estimation will serve as an estimate of the current baseline. Because range estimating was described in Section 3.5.1.2, no further explanation will be provided in this section as the proposed methodology does not seek to change the method. Furthermore, utilizing a mature methodology for the estimation of margin allows the focus of the hybrid methodology to be on forecasting prospective baseline changes and their respective effects on the initial estimate. This focus leads to Research Question 2:

Research Question 2: How can significant baseline changes resulting from phenomenological uncertainties, exogenous uncertainties, volitional uncertainties,

and certain technological uncertainties be identified and quantified?

This research question addresses two distinct analysis gaps. The first gap is the ability to identify potential baseline changes. The baseline changes identified must cover a large portion of the possibility space so that potential changes are not left out of a risk/opportunity analysis. The second gap is the ability to analyze this large number of potential baseline changes. As Section 4.1.1 shows, there can be far too many potential baselines to analyze each individually.

Before the second gap can begin to be addressed, a systematic method of identifying potential baseline changes must be created. Baseline configuration changes are the result of the underlying uncertainties, both of a technical and programmatic nature, which affect a development program. Therefore, in order to determine potential baseline changes, the uncertainties which affect the vehicle must be mapped to specific, discrete changes in the baseline configuration. The process of examining an uncertainty and determining its potential future effects has been explored in the discipline of scenario analysis.

4.2.1.1 Scenario Analysis

Scenario analysis is a forecasting technique which tries to identify a set of plausible futures known as scenarios. Scenarios are formally defined as “descriptive narratives of plausible alternative projections of a specific part of the future.” [41] Scenario analysis differs from other forecasting techniques in its ability to offer multiple forecasts. Furthermore, providing multiple forecasts is a cornerstone of scenario analysis because it allows an examination of the underlying assumptions of each forecast. [119]

Traditional scenario development begins with identifying the driving forces that can affect a project’s success. Typical categories of driving forces consist of social, economic, environmental, political, and technological factors (known as the SEEPT framework). Once the relevant driving forces have been identified, the dimensions within these driving forces which present the greatest uncertainty must be found. For example, these dimensions could encompass the price of fuel or labor as well as the projected growth of a target demographic. The two most important dimensions can then be fed into a scenario matrix. [41, 79, 119]

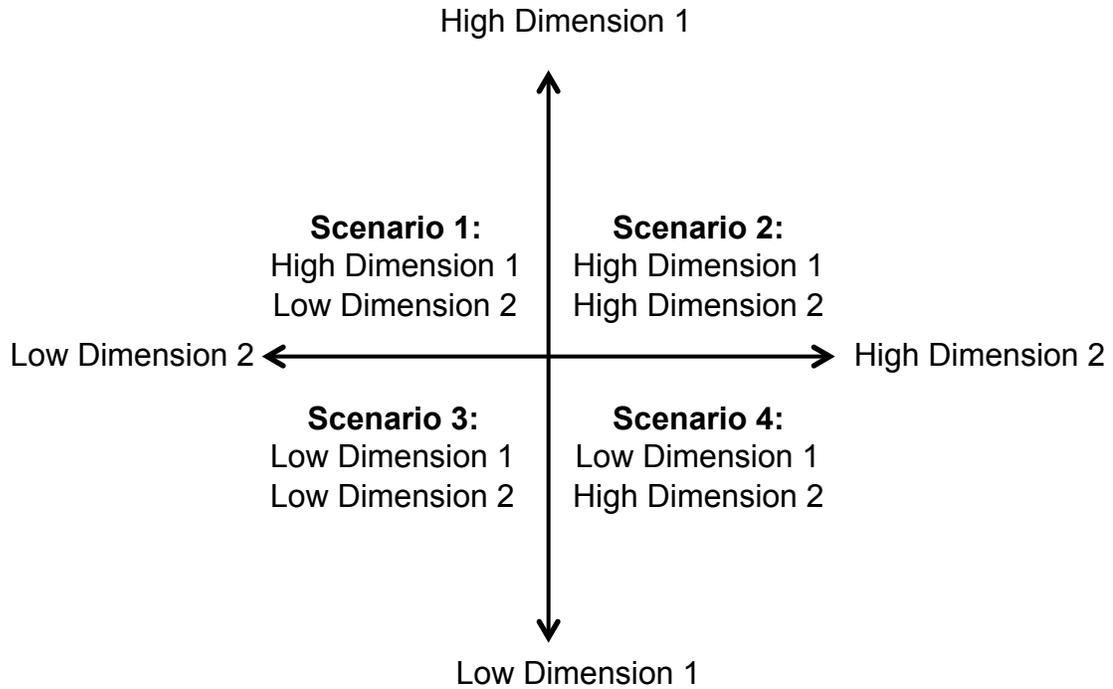


Figure 23: Sample Scenario Matrix

A scenario matrix is a 2x2 matrix which guides the creation of specific scenarios. An example scenario matrix can be seen in Figure 23. The two dimensions identified in the previous step are used as the axes of the matrix. Each quadrant of the matrix defines a separate scenario. Defining scenarios in this fashion ensures that each of the scenarios considered is qualitatively different and that the key drivers will be drivers in each of the selected scenarios. [41]

The scenario matrix results in four separate scenarios. Literature has shown that three to four scenarios is the recommended number to carry into the next phase of scenario generation, scenario plot development. [119] Scenario narrative (also known as scenario plot) development focuses on telling the story of how the world of the present turned into the world proscribed by the factors as defined in the scenario matrix. These descriptions of how today's world changes is a key feature of scenario analysis because they capture the impacts of the driving factors into an easily communicable and engaging story. This story makes the scenario more engaging for decision makers who have to consider different potential futures and consider different potential strategies. [41, 79]

4.2.1.2 Shortcomings of Scenario Analysis as Applied to Development Programs

Scenario analysis as practice lends itself towards strategic planning in large organizations. While it offers a useful method of looking at prospective futures, most of the features of this method do not apply to problem of looking at specific changes to a vehicle.

The most important feature of traditional scenario analysis, the scenario narrative, serves to tell an engaging story to communicate the implications of a potential future to decision makers. The need to develop well-defined scenario narratives drives the selection of only a few potential scenarios due to the effort required by both the scenario development team and decision makers to create and understand different visions of the future. Narratives provide decision makers in other industries the ability to understand how potential futures come about so that they can gain a better understanding of the impacts of future decisions. However, in vehicle development, the decision to determine the overall size of a vehicle and its corresponding margin can only be made once. The end-states of potential vehicle baseline changes are far more important to the eventual decision of how much margin to carry than the story documenting how those changes came about. Because this decision is only based on the scenario end-state, the narrative of how that end-state is reached is unnecessary.

Current scenario analysis treats all scenarios as equally likely— there is no assumed baseline. However, implicit in the initial design of a vehicle is a scenario where no baseline changes occur. In this baseline scenario, range estimating would account for all active uncertainties rendering a need for scenario analysis moot. A scenario development process for a vehicle will need to take the original baseline into account.

Another idea that does not carry over from traditional scenario analysis is the idea that a few key dimensions can describe the problem. Unlike large strategic planning problems where a few key driving factors can provide dimensions for a scenario matrix, a vehicle development is subject to a large number of factors which guide its future. For example, new requirements could be levied if a new stakeholder or stakeholders join the project after ATP. Technologies incorporated into the design might completely fail due to technical reasons and be removed from the vehicle, or the same technologies might prove too expensive

due to either rising costs or an externally-mandated budget reduction. Finally, as the design progresses, increased modeling fidelity may yield a result which requires a redesign.

While traditional scenario design does not address the first analysis gap identified by Research Question 2 of identifying potential baseline configuration changes, it does provide a starting point for a framework that can be specialized. By sacrificing the need to create in-depth stories focusing on how potential end-states appear, an analysis team can focus on creating more potential scenarios consisting of differing end states. This larger number of scenarios will be able to capture the uncertainty due to a larger number of driving factors.

4.2.2 Adapting Scenario-Based Forecasting to Vehicle Design

The problem of forecasting different potential baseline configurations of a vehicle requires a fundamentally different method of scenario generation and analysis. The scenario generation process begins by utilizing Kahn and Wiener’s framework for creating scenarios.

In order to make long-term projections about the future, Kahn and Wiener constructed potential scenarios of hypothetical events leading to the year 2000. The scenario construction process first involved identifying the basic trends of Western society and using these trends to create a “surprise-free” projection– the standard world. Given this initial projection, the authors then identify themes which would lead to an alternative world; from these themes eight “canonical variations” are created. These nine scenarios are meant to represent the “range of not implausible developments from existing conditions.” [66, 65]

While Kahn and Wiener were looking at a much larger problem in trying to attempt to project the state of world powers thirty years out, their basic framework for constructing scenarios is instructive to the problem of forecasting potential changes to a vehicle baseline. Kahn and Wiener created a baseline forecast and examined variations of the original standard world; a vehicle will also have a baseline forecast and potential configuration variations. With this framework identified, attention must be paid to the generation of potential alternative scenarios.

The identification of potential alternative scenarios affecting a vehicle requires harnessing the creativity and experience of the experts on a project team. Techniques for generating

potential risks/opportunities or scenarios can be as simple as pondering, defined as a person sitting down with a pen and paper and writing down potential risks or opportunities. Brainstorming is a similar process which involves a team of six to twelve individuals with a variety of backgrounds generating potential risks or opportunities. [25]

While these identified techniques are widely used, they are still potentially not exhaustive and are not structured. Literature has shown that morphological analysis is an explorative and exhaustive approach to the generation of a large number of alternatives. [63, 64, 110, 111]

4.2.2.1 Morphological Analysis

Morphological analysis was developed by Fritz Zwicky, a Swiss-American astrophysicist, in the 1940s. One of the first uses of morphology was to determine the possible design configurations of propulsion systems. [110, 153] Since this paper, morphological analysis has been used as a way to generate and collate different design solutions. [40, 94]

Morphological analysis consists of two constructs: the morphological field (also known as a morphological matrix or a matrix of alternatives) and the cross-consistency matrix. A morphological field consists of the elements of a system and delineates potential alternatives; an example morphological field can be seen in 24. Each row of the field, known as a parameter, represents a different element of the system. Members of each row, known as conditions, represent potential alternatives to an element of the system. [63, 64, 111] The selection of a condition from each parameter constitutes a single simple configuration; a simple configuration is defined as “a configuration with one and only one condition designated under each parameter.” [111] Each simple configuration represents a different alternative of the problem under consideration. The total number of simple configurations in a given morphological field is expressed by Equation 21 where n is the total number of parameters and v_i is the total number of conditions in a given parameter. This equation shows that the number of simple configurations grows geometrically with the number of parameters. [111]

$$T_{SC} = \prod_{i=1}^n v_i \quad (21)$$

RCS Propellant	NTO/MMH	LOX/Ethanol	Green Propellant	Other Green
Pitch Control	Body Flap	Canard	Both	
Tank Arrangement	Common Bulkhead	Intertank		
Tank Construction	Al-Li	Composite		
Wing Construction	Al	Al-Li	Composite	

Figure 24: Sample Morphological Field

The cross-consistency matrix is a pair-wise comparison matrix which relates conditions contained within a single parameter to the conditions in every other parameter. Each pair-wise comparison is used to determine the compatibility between conditions; if two conditions are incompatible, then no simple configurations can be made using both conditions. [63] An example cross-consistency matrix can be seen in 25; in this matrix, incompatible pairs are denoted with an 'x.' While the morphological field grows geometrically, the cross consistency matrix grows quadratically where the total number of pairs are determined by Equation 22. [111]

$$C_t = \sum_{i=1}^{n-1} \sum_{j=2}^n v_i \cdot v_j \quad (22)$$

4.2.2.2 Morphological Analysis Applied to Scenario Generation

Morphological analysis can be used to generate potential scenarios resulting in potential baseline changes. This can be used on two different levels: the generation of vehicle baseline alternatives and higher level conditions which constrain baseline alternative choices.

In this application, the first level of morphological analysis is the identification of potential ways in which the concept can be implemented. This will involve performing a physical decomposition of the vehicle and identifying alternatives for each subsystem. These alternatives can include alternate configurations, technologies, or materials.

The next level of potential alternatives concerns conditions which the project may be subject to which can result in baseline changes. Such conditions are manifestations of exogenous uncertainties; examples include funding profiles, requirements changes, and the addition or subtraction of stakeholders. A matrix of alternatives can be used to explore and

		RCS Propellant				Pitch Con.			Arrange		Tank	
		NTO/MMH	LOX/Ethanol	Green Prop	Other Green	Body Flap	Canard	Both	Common	Intertank	Al-Li	Composite
Pitch Control	Body Flap											
	Canard											
	Both											
Tank Arrangement	Com Bulk											
	Intertank											
Tank Construction	Al-Li								X			
	Composite											
Wing Construction	Al											X
	Al-Li											X
	Composite									X		

Figure 25: Sample Cross Consistency Matrix

define these conditions. While these conditions do not represent specific alterations to the vehicle baseline, they will be linked to the design alternatives matrix of alternatives through the cross-consistency matrix. The cross-consistency matrix enables the external conditions which affect the development program to explicitly define the alternatives space. Additionally, throughout this step, the morphological field representing the physical decomposition can be re-visited as new potential alternatives are identified.

These two levels define a single matrix of alternatives which describes the range of potential alternative futures. Each unique combination represents a potential future scenario. One unique scenario must define the baseline case, and, in keeping with Kahn and Wiener’s framework for scenario generation, every other alternative scenario represents a potential way for the baseline design to change. The use of a matrix of alternatives as a way to generate scenarios which define alternative baselines is formalized by Hypothesis 2.

Hypothesis 2: If a morphological approach to scenario generation is taken then large numbers of possible alternatives for the future baseline configurations can be identified.

Within a morphological matrix used for generating scenarios, each row of the matrix

Funding Profile	Cut 20%	Cut 10%	Baseline	15% Increase
Requirement 1	Baseline	Change 1	Change 2	
Requirement 2	Baseline	Change 1		
Tank Construction	Al-Li	Composite		
Wing Construction	Al	Al-Li	Composite	

Figure 26: Notional Matrix of Alternatives

represents a class of an event or design change which can occur, and each entry of a row is an alternative event or design change. A notional matrix can be seen in 26. In this example, there are 4 identified funding profiles, and two separate requirements which can be subject to change. Additionally, two structural subsystems are identified for separate construction techniques requiring differing levels of technology infusion. The baseline is identified by the green shaded alternatives.

Scenarios can play out on two different levels. The first level are the exogenous changes to the program. This is represented by differing funding profiles and potential requirements changes. A change to one of these scenario alternatives would further constrain the second level– the impacts to the vehicle. For example, if the scenario involved a funding cut of 20%, then the money needed to create composite propellant tanks will disappear. This can be modeled using the cross-consistency matrix. Once the exogenous degrees of freedom have been specified, potential changes to the baseline can still take place. For this example, if the baseline funding profiles and requirements are chosen, then the tank and wing could still have to be made out of Al-Li due to the failure of composite technology or a future trade study.

Ultimately, this hypothesis can be shown to be correct through a proof of concept. The act of creating a morphological field for a novel concept is expected to show that a sufficient number of alternatives which cover potential baseline changes can be identified.

4.2.2.3 Extending Morphological Analysis

The use of morphology to identify a large number of potential scenarios only address the first gap called to question by Research Question 2. The next challenge is the ability to analyze these scenarios quantitatively; this ability will require an extension to traditional morphological analysis to include additional information about the underlying conditions. Hypothesis 3 hypothesizes such an extension to a traditional morphological analysis.

Hypothesis 3: If a matrix of alternatives can be extended to include quantitative information about alternative baseline configurations and potential futures, then the sources of uncertainty not addressed by range estimating can be quantified.

This hypothesis serves as a frame for the required top-down analysis required by the methodology hypothesized in Hypothesis 1. This top-down methodology, known as Executable Morphological Analysis, will be developed in Chapter 5. Once this extension to morphological analysis has been developed, it will be part of a larger hybrid methodology which will be detailed in Chapter 6.

CHAPTER V

EXECUTABLE MORPHOLOGICAL ANALYSIS DATA STRUCTURES AND ALGORITHMS

As hypothesized in Section 4.2.2.3, an extension to traditional MA can be used to address the additional sources of uncertainty not addressed by range estimating. This extension allows quantitative information to be extracted from individual combinations as well as the matrix as a whole. This chapter provides an introduction to executable morphological analysis (EMA) as a general concept and develops the data structures, algorithms, and implementations specific to the problem domain of forecasting.

5.1 Previous Work into Extending Morphological Analysis

A previous study by Jimenez et al. explored extending a traditional matrix of alternatives to capture the probability and consequences of potential scenarios for a terrorist attack on a civil aviation target. Their matrix of alternatives was used to identify alternatives in the following categories: tactical objectives, point of entry, target, tool, and tool transport. In order to conduct a successful attack, an attacker will have to bypass security and carry out a technically feasible attack. To determine the probability of a successful attack, the cross-consistency matrix was populated using a [0, 1, 3, 9] scale where 0 represented no possibility of success while a 9 represents a high probability of success. The total success of a potential scenario was determined by multiplying the 10 cross-consistency matrix values together. Consequences were determined by assigning values to individual elements of the matrix, but the authors say that this is a very simple way of determining a consequence level. [64]

Applying this formulation to the problem of vehicle development is challenging because scenario elements which describe the course of a vehicle's development differ from the elements which describe a potential terrorist attack. In Jimenez et al.'s formulation, the elements which comprised an attack were not independent— a choice of one element directly

linked to the scenario through the cross-consistency matrix. However, for vehicle development, a large number of potential scenario elements are independent of one another. For example, the probability of levying a new requirement can be independent of the probability that a technology to be infused will have to be dropped from the vehicle. The presence of independent events means that probabilities cannot be determined directly through a modification of the cross-consistency matrix.

While the Jimenez et al. study shows that extending MA can be used to predict the relative probability of scenario events, it is only applicable for highly coupled problems. In order to extend MA for vehicle-centric scenario elements, a more comprehensive approach will be utilized which fundamentally changes both the morphological field and the cross-consistency matrix.

Payan's extension of MA created a probabilistic cross-consistency assessment. In this implementation, a cross-consistency assessment was performed and values between 0 and 1 were issued for each pairwise comparison. The total probability of a configuration would be the product of each pairwise comparison. This would have the effect of determining how likely a combination was to occur. [100]

5.2 Generalized Executable Morphological Analysis

The fundamental idea behind EMA is that, by including information about individual conditions, conclusions can be made about the content of the morphological field as a whole or about individual combinations of conditions. While the specific implementation of EMA will be problem-specific, there are general extensions that will be necessary for all classes of problems. First the morphological field will be extended to include multiple types of data instead of just representing potential parts of a combination. Specifically, this extension of the morphological field should include information which describes the relative likelihood and impact of selecting a condition. The inclusion of data in the morphological field will then necessitate an expansion of the cross-consistency matrix to define relationships between the morphological field's constituent conditions; these relationships will enforce compatibilities between conditions as well as influence the data contained in each condition.

Name	Parameter Alpha	A.1	A.2	A.3		
Name	Parameter Beta	B.1	B.2	B.3	B.4	B.5
Name	Parameter Gamma	Γ.1	Γ.2	Γ.3	Γ.4	

Figure 27: A Traditional Matrix of Alternatives

5.2.1 Data Structures of Executable Morphological Analysis

5.2.1.1 Executable Matrix of Alternatives

The primary data structure in traditional MA is the morphological field (also known as a matrix of alternatives). The morphological field is made up of a set of parameters; these parameters define an n-dimensional configuration space. Each parameter is in turn made up of a set of conditions. A single combination can be identified by selecting a single condition from each parameter. Figure 27 shows an example of a traditional morphological field. [111]

The executable matrix of alternatives is an extension of the morphological field for use in EMA. In order to extract quantifiable information about the combinations of the matrix, extensions need to be made on all levels of the morphological field. This section explains the generalized extension of a morphological field from Figure 27 to the executable matrix of alternatives of Figure 28.

The lowest level of the morphological field, individual conditions, define discrete choices available for selection. In order to conduct an analysis over an executable matrix of alternatives, additional information will need to be assigned to each condition. While the information necessary to add to a condition will be problem specific, at a minimum, two types of information need to be added to conditions. First, a condition will include an attribute which defines an effect, such as a weight penalty or cost increase, as a consequence of selection. Second, a condition needs to include information that can be used as an input to a function which can determine a relative likelihood or preference of one condition versus the other conditions contained in the parameter.

While extending the conditions in the morphological field is relatively straightforward ,

Name	Parameter Alpha	A.1	A.2	A.3		
Attribute 1		At1 Entry	At1 Entry	At1 Entry		
Attribute 2		At2 Entry	At2 Entry	At2 Entry		
Attribute 3		At3 Entry	At3 Entry	At3 Entry		
Name	Parameter Beta	B.1	B.2	B.3	B.4	B.5
Attribute 1		At1 Entry	At1 Entry	At1 Entry	At1 Entry	At1 Entry
Attribute 2		At2 Entry	At2 Entry	At2 Entry	At2 Entry	At2 Entry
Attribute 3		At3 Entry	At3 Entry	At3 Entry	At3 Entry	At3 Entry
Name	Parameter Gamma	Γ .1	Γ .2	Γ .3	Γ .4	
Attribute 1		At1 Entry	At1 Entry	At1 Entry	At1 Entry	
Attribute 2		At2 Entry	At2 Entry	At2 Entry	At2 Entry	
Attribute 3		At3 Entry	At3 Entry	At3 Entry	At3 Entry	

Figure 28: A Notional Executable Matrix of Alternatives

an extension to the parameters of the field is more nuanced. Within an executable matrix of alternatives, the primary function of the parameters is to provide a mechanism for selecting conditions from parent parameters. While the mechanism will be problem-specific, in general the parameter of the morphological field will have to be extended to support these mechanisms. For example, a mechanism for selecting a condition can work on the data contained within a parameter's conditions; the extension of the parameter will include constraints for ensuring that the data within the constituent conditions is well-conditioned. A specific implementation of this example would involve conditions which contain probability estimates in the form of probability estimates. In order to maintain a well-conditioned probability distribution, the parameter will have to be extended to constrain the probability values of the individual conditions so that the sum of probability values across the parameter equals unity.

Finally, algorithms will operate on the entire morphological field to extract useful information from the constituent parameters and conditions. These functions will work in concert with the extensions made to the parameters to enforce constraints on conditions during the execution of an algorithm. As with the specific implementation of parameters and conditions, the extraction algorithms will be problem dependent, but in general the

	Alpha 1	Alpha 2	Alpha 3	Beta 1	Beta 2	Beta 3	Beta 4	Beta 5
Beta 1	At 1 Relationship	At 1 Relationship	At 1 Relationship					
	At 2 Relationship	At 2 Relationship	At 2 Relationship					
	At 3 Relationship	At 3 Relationship	At 3 Relationship					
Beta 2	At 1 Relationship	At 1 Relationship	At 1 Relationship					
	At 2 Relationship	At 2 Relationship	At 2 Relationship					
	At 3 Relationship	At 3 Relationship	At 3 Relationship					
Beta 3	At 1 Relationship	At 1 Relationship	At 1 Relationship					
	At 2 Relationship	At 2 Relationship	At 2 Relationship					
	At 3 Relationship	At 3 Relationship	At 3 Relationship					
Beta 4	At 1 Relationship	At 1 Relationship	At 1 Relationship					
	At 2 Relationship	At 2 Relationship	At 2 Relationship					
	At 3 Relationship	At 3 Relationship	At 3 Relationship					
Beta 5	At 1 Relationship							
	At 2 Relationship							
	At 3 Relationship							
Gamma 1	At 1 Relationship							
	At 2 Relationship							
	At 3 Relationship							
Gamma 2	At 1 Relationship							
	At 2 Relationship							
	At 3 Relationship							
Gamma 3	At 1 Relationship							
	At 2 Relationship							
	At 3 Relationship							
Gamma 4	At 1 Relationship							
	At 2 Relationship							
	At 3 Relationship							

Figure 29: A Notional Relationship Matrix

goal of these algorithms is to extract two types of information: trends covering the overall executable matrix of alternatives and the identification of noteworthy combinations of conditions.

5.2.1.2 Relationship Matrix

The cross-consistency matrix is a pairwise comparison matrix which enforces compatibility between conditions in traditional MA. [111] This compatibility is a simple binary relationship— either conditions are compatible or not. As EMA has extended the morphological field to include more information, the cross-consistency matrix must also be extended to enable more relationships between conditions. This extension is known as the relationship matrix. A sample relationship matrix can be seen in Figure 29; this sample relationship matrix corresponds to the sample executable matrix of alternatives found in Figure 28.

The first major extension is in the way that the relationship matrix handles compatibilities. While the ability to enforce a binary relationship must remain, the relationship matrix will extend a traditional cross-consistency matrix to include continuous compatibility relationships. Relationships can make related conditions more or less likely to be

expressed. These relationships are realized by temporarily altering information contained in the executable matrix of alternatives; an activation of a relationship can change a condition's underlying data which may in turn require the enforcement of constraints on the parameter level.

In addition to continuous compatibilities, the relationship matrix can also contain relationships which change the effects of a selected condition. Relationships can make the effect of a condition more beneficial or more disadvantageous. As with compatibilities, these relationships will manifest themselves by temporarily altering the data contained within individual conditions of the executable matrix of alternatives.

The specifying of relationships in the relationship matrix data structure will be fairly straightforward. For each pairwise comparison, the relationship matrix will contain values which describe each potential relationship; a notional implementation of these relationships can be seen in Figure 29. Each pairwise comparison does not necessarily have to have a relationship which affects the values in a condition, therefore a specific implementation of relationships should have values which signify that no change should occur if a relationship is activated.

5.2.1.3 One-way Relationship Matrix

Traditional MA maintains that two conditions are either mutually compatible or incompatible. As EMA moves away from simple binary (two-way) compatibility relationships. As explained in Section 4.2.2.1, this assumption is what enables a cross-compatibility matrix to have fewer total combinations. However, it is possible that a parameter's conditions may have one-way relationships with other parameter's conditions. For example a change in the funding profile may cause designers to make different design decisions, but designers making different design decisions will not cause a change in the funding profile. Because situations like this are common, it is important for EMA to support one-way relationships.

A one-way relationship is defined as a relationship in which only one of the connected conditions can activate the relationship while only the linked condition can be affected by the relationship. Both the activating and affected conditions must be specified *a priori* during

Source Parameter	Destination Parameter	
	Condition 1	Condition 2
Condition 1		
Probability	1	1
Add Eff, Mult Rel	1	1
Add Eff, Add Rel	0	0
Mult Eff, Mult Rel	1	1
Mult Eff, Add Rel	0	0
Condition 2		
Probability	1	0.5
Add Eff, Mult Rel	0.75	1
Add Eff, Add Rel	20	50
Mult Eff, Mult Rel	9	1.2
Mult Eff, Add Rel	0.02	0.04

Figure 30: One-Way Relationship Matrix

the creation of the model. Other than the directionality of effect, one-way relationships are identical to two-way relationships for any given problem and should have the same methods of altering condition attributes and enforcing compatibilities.

The one-way relationships in a model can be visualized in a one-way relationship matrix. For each pair of parameters which can be linked through one-way relationships, an *nby*m matrix can be created where n is the number of conditions in the upper matrix parameter and m is the number of conditions in the lower matrix parameter. This is a one-way relationship matrix where the selection of a condition of the upper matrix will affect all of the conditions in the lower matrix. A one-way relationship matrix can be seen in Figure 30. In this figure, the source parameter is listed on the left, and the destination parameter is listed across the top of the matrix.

5.3 Specific Realization of EMA Data Structures

Section 5.2 introduces the general idea of EMA, its data structures, and the purposes of each extension to traditional MA. This section will apply these general ideas to an implementation of EMA specific to the problem of forecasting potential baseline configuration changes and their respective probabilities and effects.

5.3.1 Required Information in the Executable Matrix of Alternatives

5.3.1.1 *Impact of a Scenario Element*

When discussing the impact of a potential baseline change, the metric of interest is a change in mass relative to the original baseline. This can be modeled in an executable matrix of alternatives by including two attributes in each condition: an additive effect and a multiplicative effect. While other problems may have different relationships between conditions, this thesis will be focusing on using additive and multiplicative effects to model subsystem or component weights, weight deltas, and vehicle-wide weight changes.

The first effect modeled is a subsystem or component weight. In this case, the baseline condition will have an additive effect equal to the baseline weight of the subsystem or component as defined in the WBS. Alternative conditions in each parameter will also have additive effects which define the total subsystem or component weight.

Another valid implementation of additive effects in this EMA implementation can represent a change in weight relative to the subsystem or component baseline. In this case, the condition corresponding to the vehicle's baseline design will have an additive effect of 0. The conditions representing subsystem alternatives will have an additive effect equal to the delta-weight of the redesign relative to the subsystem baseline weight.

Both method of implementing additive effects can be used in the same EMA model. In general, additive effects accounting for the total weight of a component should be used when the model designer wants to expose the total weight of a subsystem to relationships. Additive effects which only account for delta-weights should be used when representing components which are not likely to be affected by other parts of the model.

While a scalar value can account for most scenarios, certain scenarios will affect the entire vehicle necessitating a percentage change in mass. For example, implementing a vehicle-wide mass control program can decrease the overall mass of the vehicle compared to a development program which foregoes a mass control program. These changes can be implemented through a multiplicative effect: a scalar multiplication of the vehicle's total mass.

For a given combination of conditions, the total additive effect is equal to the sum of

the additive effects in each selected condition; this formula is expressed in Equation 23. Similarly, the total multiplicative effect for a given combination of conditions is equal to the product of all multiplicative effects; this is expressed in Equation 24. It is expected that very few conditions will result in the scalar multiplication of the vehicle mass; for all conditions which do not have a multiplicative effect, the value of the multiplicative effect should be assigned as unity so that it will not affect the output of Equation 24.

$$E_{additive_{combination}} = \sum_{i=1}^n E_{additive_i} \quad (23)$$

$$E_{multiplicative_{combination}} = \prod_{i=1}^n E_{multiplicative_i} \quad (24)$$

Once the total additive and multiplicative effects are known, the next step is to calculate the total weight of the vehicle. Because additive effects can either represent the baseline weight or a delta-weight, the remaining weight not accounted for by the EMA model must be calculated. This remaining weight is calculated according to Equation 25 where the remaining weight is equal to the WBS baseline weight minus the sum total of all baseline conditions in the EMA model. Given this remainder, the total weight of the vehicle accounting for both additive and multiplicative effects is calculated according to Equation 26.

$$Weight_{remaining} = Weight_{baseline} - \sum_{i=1}^n E_{additive_{baseline_i}} \quad (25)$$

$$Weight_{total} = (Weight_{remaining} + E_{additive_{combination}}) * E_{multiplicative_{combination}} \quad (26)$$

The output of the EMA model and Equation 26 provides a forecast for the total weight of the vehicle. In order to extract a design margin from this forecast, the baseline weight as defined by the WBS must be subtracted from the EMA model output; this relationship is expressed in Equation 27.

$$Margin_{Vehicle} = Weight_{ModelOutput} - Weight_{Baseline} \quad (27)$$

These equations fully explain how to calculate the total weight and margin for a single-item weight statement, but most problems will have a hierarchal weight statement broken down into subsystems and components. In EMA, each parameter can be assigned to a particular level of a weight statement in order to correspond with a vehicle's WBS.

For a selection of conditions in parameters which correspond to a particular subsystem or component, the total additive effect is calculated using Equation 28; the total additive effect for the subsystem or component is simply the sum of the additive effects of the subsystem's constituent parameters. Additionally, subsystems are not affected by multiplicative effects as these multiply the total vehicle baseline and are designed to model vehicle-wide influences; subsystem-specific multiplicative effects can be modeled through relationships as described in Section 5.3.2.2.

$$E_{additive_{subsystem}} = \sum_{i=1}^j E_{additive_i} \quad (28)$$

In order to calculate the total weight of a subsystem, the remaining weight not accounted for by the EMA model must be calculated. This remaining weight is calculated using Equation 29; this process is the same as for a single-line WBS except that this would be repeated for each subsystem modeled. The total subsystem weight can be calculated using Equation 30. Equation 30 shows that, unlike the total vehicle weight case, the total subsystem weight is only a function of additive effects. Finally, the total subsystem margin can be calculated by subtracting the subsystem baseline weight from the model output as calculated by Equation 30; this margin calculation is shown in Equation 31.

$$Weight_{remaining_{subsystem}} = Weight_{baseline_{subsystem}} - \sum_{i=1}^j E_{additive_{baseline_i}} \quad (29)$$

$$Weight_{total_{subsystem}} = Weight_{remaining_{subsystem}} + E_{additive_{subsystem}} \quad (30)$$

$$Margin_{Subsystem} = Weight_{ModelOutput_{Subsystem}} - Weight_{baseline_{subsystem}} \quad (31)$$

Because a WBS is hierarchal, the total weight of a system will be equal to the sum of

its constituent subsystems. This hierarchal relationship must also hold for the output of an EMA model; as can be seen in Equation 32, the weight of a system is equal to the sum of j subsystem weights plus k conditions which only affect the higher-level system and the remaining weight of the system. The remaining weight of a system is equal to the WBS baseline weight minus the baseline weights of all constituent subsystems and all parameters attributed to the system; this is shown in Equation 33 where there are j subsystems and k system conditions. This remaining weight will account for any constituent subsystems not specifically modeled through EMA. If all of a system's subsystems are accounted for in the EMA model, then the system's remaining weight should equal 0, and there should be no conditions which only affect the system level weight statement.

$$Weight_{system} = \sum_{i=1}^j Weight_{total_subsystem} + \sum_{i=1}^k E_{additive_i} + Weight_{remaining_system} \quad (32)$$

$$Weight_{remaining_system} = Weight_{baseline_system} - \left(\sum_{i=1}^j Weight_{baseline_i} + \sum_{i=1}^k E_{additive_baseline_k} \right) \quad (33)$$

When the system represents the top level of a WBS, Equation 32 will calculate the ($Weight_{remaining} + E_{additive_combination}$) portion of Equation 26. At this point the multiplicative effects on the total vehicle should be applied; the added weight due to the multiplicative effects can be accounted for by an additional high-level line on a WBS unattributable to a specific subsystem.

5.3.1.2 Probability of a Scenario Element

While a condition can contain a number of attributes which describe a scenario, the attribute or attributes which can be used to calculate a fitness or scenario probability are more critical to the goal of determining which combinations are most likely to occur. For this problem, each condition will have a field which specifies its probability of occurring relative to other conditions in the same parameter. The direct specification of a probability for conditions within a parameter was selected for two reasons: it is an explicit measure of the relative

Alpha	A1	A2	A3	A4
Prob	0.25	0.5	0.1	0.15
Beta	B1	B2		
Prob	0.75	0.25		

Figure 31: Example Matrix with Probabilities

Table 10: Joint Probability Distribution for Example Matrix with Probabilities

	A1	A2	A3	A4	Row total
B1	0.1875	0.375	0.075	0.1125	0.75
B2	0.0625	0.125	0.025	0.0375	0.25
Col Total	0.25	0.5	0.1	0.15	1

probabilities, and probability estimates can be elicited using techniques from the field of subjective probability forecasting.

An explicit measure of probability is very useful as it enables the direct calculation of statistics; the expected value of the change in mass and percentiles of potential mass estimates will be available to decision makers. This direct specification is also subject to constraints on potential values. First of all, all probability values must be ≥ 0 . Additionally, a constraint will need to be placed on the individual conditions of a parameter; the sum total of the probabilities in a parameter's constituent conditions must sum to unity. By ensuring that the probabilities of the conditions in each parameter sum to unity, the total matrix represents n jointly distributed discrete random variables where n is equal to the number of parameters. Therefore, the specification of probability values for a given parameter is directly specifying the marginal distribution of a parameter. [58, 141]

An example probability matrix can be seen in Figure 31. As this represents two jointly distributed discrete random variables, the joint probability distribution can be seen in Table 10. For the basic definition of probabilities, each parameter can be modeled as an independent variable. Therefore, the generalized calculation for the combined probability for a combination of condition selections is expressed by Equation 34; the total probability is equal to the product of the combination's constituent probability values.

$$P_{combination} = \prod_{i=1}^n P_i \quad (34)$$

An additional consideration in selecting the direct specification of probability is the act of eliciting a value during the construction of this data structure. In order to assign a specific probability value, the experts consulted to make an elicitation will have to decide on a specific number to assign for a probability— a common problem in the fields of risk analysis and forecasting. Because it is a common problem, this is an active area of research with several documented methods of eliciting subjective probability estimates.[52, ?, 86, 88] An organization employing EMA would be able to choose a suitable method based on their experience and organizational needs.

5.3.2 Relationships

To complete the realization of EMA for the purposes of forecasting potential vehicle baseline changes, the relationships which comprise the model need to fully describe the potential relationships between individual conditions and their constituent data elements. This will involve three types of relationships: relationships affecting probability values, relationships affecting additive effects, and relationships affecting multiplicative effects. Additive and multiplicative effects can be modified through either additive or multiplicative relationships; this leads to a total of five potential relationships possible in an EMA model.

Pairs of conditions may be linked by all, none, or any combination of the five relationships described in this section. If two conditions are not linked via a relationship, then the values in the relationship matrix or one-way relationship matrix are defaulted to designated “no effect” values.

5.3.2.1 Relationships Affecting Probabilities

The probabilities enumerated in Section 5.3.1.2 specify the probability of a condition being selected relative to the other conditions in a parameter. Another way of looking at the relative probabilities within a parameter is to examine at the ratios of probabilities. For the matrix in Figure 32, the ratios of each pair of probabilities can be seen in Table 11.

Alpha	A1	A2	A3	A4
Prob	0.25	0.5	0.1	0.15
Beta	B1	B2	B3	
Prob	0.6	0.3	0.1	

Figure 32: Expanded Example Matrix with Probabilities

Table 11: Ratios of Probabilities for Figure 32

A1/A2	A1/A3	A1/A4	A2/A3	A2/A4	A3/A4
1/2	5/2	5/2	5	10/3	2/3
B1/B2	B1/B3	B2/B3			
2	6	3			

These ratios of probabilities are of primary importance when considering relationships because a straightforward way to express a probability relationship is to specify the change in probability relative to every other condition in the parameter.

Probability relationships between conditions are defined by Equation 35. In this equation, $A_{selected}$ represents the selected condition from parameter A, B_{target} represents the adjoining condition primarily affected by the relationship, and B_{other} represents other conditions in parameter B. $R_{Relationship}$ is a multiplicative relationship between ratios of probabilities.

$$\frac{P(B_{adjoining})|A_{selected}}{P(B_{other})|A_{selected}} = R_{relationship} * \frac{P(B_{adjoining})}{P(B_{other})} \quad (35)$$

These probability relationships are implemented in two phases. The first phase is to multiply the specified probability value in the adjoining condition by the relationship value. This first step is expressed in Equation 36; if two conditions do not have a probability relationship, then $R_{relationship}$ in this equation will be equal to unity. Once this step has been carried out for each relationship between the selected condition and the adjoining parameter, the conditions of the adjoining parameter are normalized so that the sum of all probability values is equal to unity. This normalization process ensures that Equation 35 holds for all probability ratios.

Alpha	A1	A2	A3	A4
Prob	0.25	0.5	0.1	0.15
Beta	B1	B2	B3	
Prob	0.6	0.6	0.3	

Figure 33: Expanded Example Matrix After Selection of Condition A2

Alpha	A1	A2	A3	A4
Prob	0.25	0.5	0.1	0.15
Beta	B1	B2	B3	
Prob	0.4	0.4	0.2	

Figure 34: Expanded Example Matrix After Normalization of Parameter B

$$P_{modified} = P_{specified} * R_{relationship} \quad (36)$$

$$P_{new} = \frac{P_{modified}}{\sum P_{modified_i}} \quad (37)$$

For the matrix specified in Figure 32 with probability ratios defined by Table 11, the condition A2 has relationships with B2 and B3 such that the posterior probabilities of B2 and B3 result in the conditions being twice and three times as likely to be chosen given the selection of A2. Figure 33 shows the state of the matrix immediately after the selection of A2: $P(B2)_{intermediate} = 2 * P(B2)$ and $P(B3)_{intermediate} = 3 * P(B3)$. The next step is to apply Equation 37 to parameter B; this change can be seen in Figure 34. The ratios of probabilities for this matrix can be seen in Table 12. These new values match the definition of probability relationships as defined by Equation 35: B1/B2 is now equal to 1, B1/B3 is now equal to 2, and B2/B3 is now equal to 2.

Table 12: Ratios of Probabilities for Figure 34

A1/A2	A1/A3	A1/A4	A2/A3	A2/A4	A3/A4
0.5	5/2	5/2	5	10/3	2/3
B1/B2	B1/B3	B2/B3			
1	2	2			

Alpha	A1	A2	A3	A4
Prob	0.25	0.5	0.1	0.15
Beta	B1	B2	B3	
Prob	0	0.3	0.1	

Figure 35: Expanded Example Matrix After Selection of Condition A2

Alpha	A1	A2	A3	A4
Prob	0.25	0.5	0.1	0.15
Beta	B1	B2	B3	
Prob	0	0.75	0.25	

Figure 36: Expanded Example Matrix After Normalization of Parameter B

These probability relationships can also be used to implement incompatibilities within an extended morphological field and maintain the functionality of the cross-consistency matrix. For two conditions to be incompatible, the probability relationship linking them should be equal to 0. This will result in the adjoining condition’s probability being equal to 0 after the application of Equation 36. For example, suppose that A2 and B1 from the matrix shown in Figure 32 are incompatible. This will result in B1’s probability value being equal to 0 as seen in Figure 35. After the normalization of parameter B, the final state of the matrix is as seen in Figure 36, and the corresponding probability ratios are shown in Table 13. This table shows that this relationship renders A2 and B1 incompatible while maintaining the original probability ratio between B2 and B3.

Each probability relationship can only be applied once during the process of selection of an extended morphological field. Once a condition is selected from a parameter, the probability values for the conditions in that parameter are fixed for the remainder of the

Table 13: Ratios of Probabilities for Figure 36

A1/A2	A1/A3	A1/A4	A2/A3	A2/A4	A3/A4
0.5	5/2	5/2	5	10/3	2/3
B1/B2	B1/B3	B2/B3			
0	0	3			

selection process. This ensures that Equation 34 accurately reflects the overall probability of selecting a series of conditions based on the order in which they are selected.

5.3.2.2 Relationships Affecting Condition Attributes

Conceptually, relationships between conditions which affect changes in the expected mass are much simpler. The expected mass can be linked through either a multiplicative or additive relationship. However, these relationships can be more complicated to implement because there are more potential relationships between effects. Both multiplicative and additive effects on the overall mass of the vehicle can be affected by additive relationships, multiplicative relationships, or both additive and multiplicative relationships.

Additive effects for a change in mass can be modified by an additive relationship, a multiplicative relationship, or both additive and multiplicative relationships. The equation for these relationships can be seen in Equation 38. In this equation, the adjoining condition's new effect value is modified by both the multiplicative and additive relationship. Both additive and multiplicative relationships can be expressed by the same equation because the default relationship values (0 for additive relationships and 1 for multiplicative relationships) will result in no change to condition attributes. Furthermore, this equation holds for both one-way relationships and two-way relationships.

$$E_{Additive_{new}} = (E_{Additive_{specified}} * M_{relationship}) + A_{relationship} \quad (38)$$

Multiplicative relationships affecting additive effects are more complicated because the relationship needs to act on both the selected and adjoining condition in order to maintain consistency. For example, if condition 1 has an impact of 300 lbs and condition 2 has an impact of 400 lbs, a multiplicative relationship resulting in an additional 20% growth could result in an extra 60 or 80 lbs added to the overall mass depending on which condition was evaluated with the multiplicative relationship. This inconsistency can be avoided by making the multiplicative relationship propagate backwards upon selection of the original adjoining condition. When the original adjoining condition is selected, the original selected condition's additive effect is multiplied by the multiplicative relationship; this is expressed

in Equation 39. This equation will only be active in two-way relationships.

$$E_{Additive} = E_{Additive_{specified}} * M_{relationship} \quad (39)$$

In addition to additive effects, this implementation of EMA can also have multiplicative effects which can in turn be subject to both additive and multiplicative relationships. Equation 40 shows the expression for both additive and multiplicative relationships affecting multiplicative effects. As with relationships affecting additive effects, relationships affecting multiplicative effects can be expressed through a single equation because the default relationship values (0 for additive relationships and 1 for multiplicative relationships) will result in no change to condition attributes. This equation shows the forward propagation of a relationship and holds for both one-way relationships and two-way relationships.

$$E_{Multiplicative_{new}} = (E_{Multiplicative_{specified}} * M_{relationship}) + A_{relationship} \quad (40)$$

For the case of additive relationships acting on a multiplicative effect, the additive relationship should act on both the selected and adjoining conditions. This is the opposite of the way that relationships affecting additive effects are handled because multiplicative effects are subject to Equation 24 and multiplied together at the end of selection. The backward propagation of additive relationships affecting multiplicative effects is expressed in Equation 41. This evaluation of this equation is triggered when the original adjoining condition is selected during the execution of an EMA model; upon selection, the originally selected condition's multiplicative effect will be modified by the additive relationship. Furthermore, this equation only holds for two-way relationships defined in the relationship matrix.

$$E_{Multiplicative_{new}} = E_{Multiplicative_{specified}} + A_{relationship} \quad (41)$$

5.4 *Specific Realization of EMA Algorithms*

Section 5.2 talks about the need to implement problem-specific algorithms for extracting information from EMA data structures as defined in Section 5.3. Specifically, algorithms

will be used to quantify the effect and probability data contained in these data structures; furthermore this quantification should take the form of a probability distribution so that the results can synerize with bottom-up estimates made by range estimating. In addition to working directly with range estimating, search algorithms will be needed to extract individual combinations which are of interest to designers– the most likely cases, adverse conditions, or a combination of high probability and adversity.

This section covers all aspects of extracting information from EMA data structures. First, the act of selecting a single combination, an act common to both search and probability distribution generating algorithms, is discussed. Next, the direct calculation of a probability mass function from an executable matrix of alternatives is shown; this leads to Experiment 1 which shows that a Monte Carlo simulation is a useful method of constructing a continual distribution function of the information contained in an executable matrix of alternatives. The final section discusses a search algorithm which can find interesting combinations in the executable matrix of alternatives.

5.4.1 Selecting a Combination

Before one can synthesize an algorithm to extract trends from the executable matrix of alternatives as a whole, the process for selecting a single potential alternative must be discussed. This section walks through a sample selection of a group of conditions in order to determine how to calculate a combination’s total probability and total effect.

For the purposes of this example, the sample executable matrix of alternatives can be found in Figure 37. The corresponding relationship matrix can be found in Figure 38. This relationship matrix only contains additive and multiplicative relationships which affect additive effects; the relationships which affect multiplicative effects are not listed in Figure 38 because they are trivial and express no change in effect. This example will only discuss updates to the probabilities and the additive effects of the members of Figure 37.

Name		A.1	A.2	A.3		
Mult Effect	Parameter	1	1	1		
Add Effect	Alpha	400	0	300		
Probability		0.2	0.5	0.3		
Name		B.1	B.2	B.3	B.4	B.5
Mult Effect	Parameter	1.1	1.2	0.9	1	1
Add Effect	Beta	-300	500	150	250	0
Probability		0.15	0.15	0.2	0.2	0.3
Name		Γ.1	Γ.2	Γ.3	Γ.4	
Mult Effect	Parameter	1	1	1	1	
Add Effect	Gamma	0	500	700	-300	
Probability		0.4	0.4	0.1	0.1	
Name		Δ.1	Δ.2			
Mult Effect	Parameter	1	1			
Add Effect	Delta	0	700			
Probability		0.6	0.4			
Name		E.1	E.2	E.3		
Mult Effect	Parameter	1	1	1		
Add Effect	Epsilon	1000	500	0		
Probability		0.3	0.3	0.4		

Figure 37: Executable Matrix of Alternatives used for Selection Example

		A.1	A.2	A.3	B.1	B.2	B.3	B.4	B.5	Г.1	Г.2	Г.3	Г.4	Δ.1	Δ.2
Add. Rel.	B.1	0	-100	0											
Mult. Rel		1	1	1											
Prob		1	0.9	1											
Add. Rel.	B.2	0	0	0											
Mult. Rel		1	1	1											
Prob		1	0	1											
Add. Rel.	B.3	0	0	0											
Mult. Rel		1	1	1											
Prob		1	1	1											
Add. Rel.	B.4	100	100	50											
Mult. Rel		1	1	1											
Prob		1.2	0.9	1											
Add. Rel.	B.5	0	0	0											
Mult. Rel		1	1	1											
Prob		1	1	1											
Add. Rel.	Г.1	0	0	0	0	0	0	0	0						
Mult. Rel		1	1	1	1	1	1	1	1						
Prob		1	1	1	1	1	1	1	1						
Add. Rel.	Г.2	0	0	0	0	-50	0	0	0						
Mult. Rel		1	1	1	1	1	1	1	1						
Prob		1	1	0	1	1.2	1	1	1						
Add. Rel.	Г.3	0	0	0	0	0	0	-200	0						
Mult. Rel		1	1	1	1	1	1	1	1						
Prob		1	1	1	1	1	1	1.5	1						
Add. Rel.	Г.4	-150	0	0	150	0	0	0	0						
Mult. Rel		1	1	1	1	1	1	1	1						
Prob		0.75	1	1	1.2	1	1	1	1						
Add. Rel.	Δ.1	130	0	0	0	0	0	0	0	0	0	0	0		
Mult. Rel		1	1	1	1	1	1	1	1	1	1	1	1		
Prob		1.75	1	1	1	0	1	1	1	1	1	1	1		
Add. Rel.	Δ.2	0	0	-50	0	0	0	0	0	0	0	0	0		
Mult. Rel		1	1	1	1	1	1	1	1	1	1	1	1		
Prob		1	1	0.9	1	1	1	1	1	1	1	1	1		
Add. Rel.	E.1	75	-50	70	200	-300	40	-200	-30	75	50	-150	-250	415	-200
Mult. Rel		1	1	1	1	1	1	1	1	1	1	1	1	1.5	0.5
Prob		1.2	1	1	1	0.5	1	0.5	0	1	1	0.75	0.9	1.2	0.85
Add. Rel.	E.2	65	150	200	-300	75	30	125	-225	-140	-75	-100	45	-75	125
Mult. Rel		1	1	1	1	1	1	1	1	1	1	1	1	0.75	3
Prob		1	1	1	1	1	1	1	1	0	1	1.2	1	1	1
Add. Rel.	E.3	-85	125	-35	90	-200	140	-65	80	150	25	85	-65	-180	130
Mult. Rel		1	1	1	1	1	1	1	1	1	1	1	1	2	1.2
Prob		1	0	1	1	1	1	1	1	1	1	1	1	1	1

Figure 38: Relationship Matrix used for Selection Example

The selection of a specific combination from an executable matrix of alternatives utilizes the equations derived in Section 5.3.2, specifically Equations 36, 37, 38 and 39. In this example the following conditions are selected from Figure 37 in order: A1, B4, Γ 4, Δ 1, and E1.

The first condition selected from this matrix is condition A1. A1 has relationships with B4, Γ 4, Δ 1, and E1. Figure 39 shows the consequences of selecting A1 where B4, Γ 4, Δ 1, E1, and Z1 have all been updated through the application of Equations 36 and 38 to the original values and relationships found in Figure 38. After the application of relationships, the probability values for each parameter violate the constraint that the sum of probabilities across a parameter must equal one. In order to address this violation, Equation 37 is applied to each parameter; the resulting executable matrix of alternatives after a normalization can be seen in Figure 40.

These two figures illustrate the application of additive relationships to the additive effect as well as the probability relationships which change the relative probabilities within a parameter. Because the selections of B4 and Γ 4 also only involve additive relationships, the example will continue after their selections, but before the selection of Δ 1 which involves a multiplicative relationship with E1. The state of the executable matrix of alternatives before the selection of Δ 1 can be seen in Figure 41.

Figure 42 shows the state of the matrix after the selection of Δ 1 and the normalization of each parameter's probabilities. The additive effects of E1, E2, and E3 are modified by their multiplicative relationships with Δ 1 through equation 38.

After E1 is selected, Equation 39 is applied. This triggers the multiplicative relationship with Δ 1 and Δ 2. The final state of the extended matrix after the selection of condition E1 can be seen in Figure 43. This illustrates the idea developed in Section 5.3.2.2 that a multiplicative relationship will affect both of the conditions linked through a multiplicative relationship.

After the individual conditions are selected, the overall probability and effect for the combination can be calculated. The overall probability of a combination is the product of the individual probabilities of the combinations constituent conditions (the multiplicative

Name		A.1	A.2	A.3		
Mult Effect	Parameter	1	1	1		
Add Effect	Alpha	400	0	300		
Probability		0.2	0.5	0.3		
Name		B.1	B.2	B.3	B.4	B.5
Mult Effect	Parameter	1.1	1.2	0.9	1	1
Add Effect	Beta	-300	500	150	350	0
Probability		0.15	0.15	0.2	0.24	0.3
Name		Γ .1	Γ .2	Γ .3	Γ .4	
Mult Effect	Parameter	1	1	1	1	
Add Effect	Gamma	0	500	700	-450	
Probability		0.4	0.4	0.1	0.075	
Name		Δ .1	Δ .2			
Mult Effect	Parameter	1	1			
Add Effect	Delta	130	700			
Probability		1.05	0.4			
Name		E.1	E.2	E.3		
Mult Effect	Parameter	1	1	1		
Add Effect	Epsilon	1075	500	0		
Probability		0.36	0.3	0.4		

Figure 39: The Selection of Condition A1 and the Activation of Relationships

Name		A.1	A.2	A.3		
Mult Effect	Parameter	1	1	1		
Add Effect	Alpha	400	0	300		
Probability		0.2	0.5	0.3		
Name		B.1	B.2	B.3	B.4	B.5
Mult Effect	Parameter	1.1	1.2	0.9	1	1
Add Effect	Beta	-300	500	150	350	0
Probability		0.1442	0.1442	0.1923	0.2308	0.2885
Name		Γ .1	Γ .2	Γ .3	Γ .4	
Mult Effect	Parameter	1	1	1	1	
Add Effect	Gamma	0	500	700	-450	
Probability		0.4103	0.4103	0.1026	0.0769	
Name		Δ .1	Δ .2			
Mult Effect	Parameter	1	1			
Add Effect	Delta	130	700			
Probability		0.7241	0.2759			
Name		E.1	E.2	E.3		
Mult Effect	Parameter	1	1	1		
Add Effect	Epsilon	1075	565	-85		
Probability		0.3396	0.283	0.3774		

Figure 40: The Selection of Condition A1 and the Normalization of Probabilities

Name		A.1	A.2	A.3		
Mult Effect	Parameter	1	1	1		
Add Effect	Alpha	400	100	350		
Probability		0.2	0.5	0.3		
Name		B.1	B.2	B.3	B.4	B.5
Mult Effect	Parameter	1.1	1.2	0.9	1	1
Add Effect	Beta	-150	500	150	350	0
Probability		0.1442	0.1442	0.1923	0.2308	0.2885
Name		Γ.1	Γ.2	Γ.3	Γ.4	
Mult Effect	Parameter	1	1	1	1	
Add Effect	Gamma	0	500	500	-450	
Probability		0.3902	0.3902	0.1463	0.0732	
Name		Δ.1	Δ.2			
Mult Effect	Parameter	1	1			
Add Effect	Delta	130	700			
Probability		0.7241	0.2759			
Name		E.1	E.2	E.3		
Mult Effect	Parameter	1	1	1		
Add Effect	Epsilon	625	735	-215		
Probability		0.2358	0.3275	0.4367		

Figure 41: The State of the Executable Matrix of Alternatives after the selections of A1, B4, and Γ4

Name		A.1	A.2	A.3		
Mult Effect	Parameter	1	1	1		
Add Effect	Alpha	400	100	350		
Probability		0.2	0.5	0.3		
Name		B.1	B.2	B.3	B.4	B.5
Mult Effect	Parameter	1.1	1.2	0.9	1	1
Add Effect	Beta	-150	500	150	350	0
Probability		0.1442	0.1442	0.1923	0.2308	0.2885
Name		Γ .1	Γ .2	Γ .3	Γ .4	
Mult Effect	Parameter	1	1	1	1	
Add Effect	Gamma	0	500	500	-450	
Probability		0.3902	0.3902	0.1463	0.0732	
Name		Δ .1	Δ .2			
Mult Effect	Parameter	1	1			
Add Effect	Delta	130	700			
Probability		0.7241	0.2759			
Name		E.1	E.2	E.3		
Mult Effect	Parameter	1	1	1		
Add Effect	Epsilon	1352	476.25	-610		
Probability		0.2702	0.3128	0.417		

Figure 42: The State of the Executable Matrix of Alternatives after the selections of A1, B4, Γ 4, and Δ 1

Name		A.1	A.2	A.3		
Mult Effect	Parameter	1	1	1		
Add Effect	Alpha	400	100	350		
Probability		0.2	0.5	0.3		
Name		B.1	B.2	B.3	B.4	B.5
Mult Effect	Parameter	1.1	1.2	0.9	1	1
Add Effect	Beta	-150	200	150	350	0
Probability		0.1442	0.1442	0.1923	0.2308	0.2885
Name		Γ.1	Γ.2	Γ.3	Γ.4	
Mult Effect	Parameter	1	1	1	1	
Add Effect	Gamma	0	500	500	-450	
Probability		0.3902	0.3902	0.1463	0.0732	
Name		Δ.1	Δ.2			
Mult Effect	Parameter	1	1			
Add Effect	Delta	195	150			
Probability		0.7241	0.2759			
Name		E.1	E.2	E.3		
Mult Effect	Parameter	1	1	1		
Add Effect	Epsilon	1352	476.25	-610		
Probability		0.2702	0.3128	0.417		

Figure 43: The State of the Executable Matrix of Alternatives after the selections of A1, B4, Γ4, Δ1, and E1

rule [58, 141]); this relationship is expressed in Equation 34 where n is equal to the number of parameters in the morphological field. The additive and multiplicative effects will be summed and multiplied respectively to calculate the overall effects for the combination as expressed in Equations 23 and 24. Finally, the overall effect on the baseline system is achieved by adding the additive effect and multiplying that sum with the multiplicative effect; this is expressed in Equation 26. This example selection will have a probability of .000661, a multiplicative effect of unity, and a total additive effect of 1,847.

In order to illustrate the use of subsystems, assume that parameters α and γ are assigned to subsystem 1, and δ and ϵ are assigned to subsystem 2. The total weight of each subsystem would be calculated according to Equation 28 because the remaining weight value in Equation 30 is equal to 0. In this case subsystem 1 (α and γ) has a total weight effect of -50 lbs, and subsystem 2 (δ and ϵ) has a total weight effect of 1,547 lbs. Equations 32 and 33 can be applied to calculate the total weight of the system. In this case, the remaining weight is equal to 350, the value of $B.4$. This leads to a total system weight of 1847, equal to the total additive effect for the full matrix. This value would then be multiplied by the total multiplicative effect for the matrix which in this example is equal to unity.

5.4.2 Potential Options

While Section 5.4.1 examines the selection of a single combination of a matrix, the evaluation of the entire matrix to generate trends is required in order to make a forecast. An algorithm which evaluates the entire matrix will need to provide utility for decision makers and synergize with the results of range estimating.

Previous studies into probabilistic design and margin estimation have shown that the cumulative distribution function (CDF) is a key function which can be used by decision makers. [31, 132] The focus of an algorithm used to extract data from the executable matrix of alternatives should be on generating a CDF, or a probability mass function (PMF) which could be integrated to generate a CDF. [58, 141]

The first potential method of generating a PMF would be from the evaluation of every

potential alternative in the executable matrix of alternatives. This would involve cycling through every potential combination and following the selection process as outlined in Section 5.4.1. As each selection would yield a feasible alternative with both a probability and an effect, the totality of all potential selections would yield a PMF. This PMF could then be integrated to form a CDF.

Another potential option would be performing a Monte Carlo simulation over the executable matrix of alternatives. This Monte Carlo simulation would randomly select combinations based on the probability values contained within the individual combinations. These random selections could then be used to construct an empirical CDF.

The presence of two options necessitates a choice in how to proceed forward with the algorithm to extract probabilistic information from the executable matrix of alternatives. A direct calculation of the PMF would yield a very accurate calculation, but Equation 21 shows that the number of potential combinations grows at an extremely fast rate. Previous studies have also shown that morphological analysis can create extremely large numbers of potential combinations to the point where comprehensive analysis would be impractical. [40] The combinatorial problem in EMA is further exacerbated because the order in which conditions are chosen from the executable matrix of alternatives will create multiple different combination outcomes compared to the analogous case in traditional MA which would only yield a single combination. Because of the possibility of an untenable number of potential combinations, the preferred algorithm should be Monte Carlo simulation. This is expressed in Hypothesis 3a.

Hypothesis 3a: A Monte Carlo simulation run over an Executable Matrix of Alternatives will yield an equilibrium distribution which describes the impact of the alternative combinations derived from the morphological field. This equilibrium distribution will require less computational resources to compute compared to a total sum of all potential combinations.

This hypothesis will be approached from two sets of analyses. First, the properties of the executable matrix of alternatives will be examined. This section will explain how a PMF can be directly calculated from an executable matrix of alternatives, and it will examine the necessary computational requirements for this calculation. In comparison,

an experiment will be performed to determine how many Monte Carlo simulation runs are necessary to find an equilibrium distribution based on the executable matrix of alternatives. If the computational requirements for CDF generation via Monte Carlo simulation are less strenuous than those required for the direct calculation of the PMF, then the hypothesis is accepted.

5.4.3 Direct Calculation of PMF

The most basic option for calculating the PMF of an executable matrix of alternatives would be the evaluation of every possible combination of conditions. In order for the direct calculation of a PMF to work, combinations must generate a specific effect and probability, and the executable matrix of alternatives must behave as n jointly distributed discrete random variables where n is the number of parameters in the matrix.

An executable matrix of alternatives will act as a discrete joint probability distribution if the probability values follow the rules as laid out in Equation 42. [58] Rule number one is enforced by the constraint that individual probability values must be ≥ 0 ; if all probability values are ≥ 0 then the product of probability values will also be ≥ 0 . Rule number three is satisfied because Equation 34 defines the probability of a combination as a function of the selections of a combination.

1. $f(x, \dots, n) \geq 0$ for all (x, \dots, n)
2. $\sum_x \dots \sum_n f(x, \dots, n) = 1$ (42)
3. $P(X = x, \dots, N = n) = f(x, \dots, n)$

Rule number 2, the summation of all discrete probabilities to unity, is dependent on the definition of the executable matrix of alternatives. For the simplest implementations, a relationship matrix can omit all relationships between the probabilities of individual conditions. In this case, the executable matrix of alternatives can be thought of as n independent variables where n is the number of parameters in the matrix. In this case, the total probability that a combination of conditions will occur will follow the multiplicative rule, Equation 34. [58, 141] As the constituent conditions of each parameter are constrained to sum to unity, the sum of every combination of conditions will also sum to unity.

Alpha	A1	A2	A3	A4
Prob	0.25	0.5	0.1	0.15
Beta	B1	B2	B3	
Prob	0.6	0.3	0.1	

Alpha	A1	A2	A3	A4
Prob	0.25	0.5	0.1	0.15
Beta	B1	B2	B3	
Prob	0.6	0.6	0.1	

Alpha	A1	A2	A3	A4
Prob	0.25	0.5	0.1	0.15
Beta	B1	B2	B3	
Prob	0.461538	0.461538	0.076923	

Alpha	A1	A2	A3	A4
Prob	0.25	0.5	0.1	0.15
Beta	B1	B2	B3	
Prob	0.461538	0.461538	0.076923	

Alpha	A1	A2	A3	A4
Prob	0.25	1	0.1	0.15
Beta	B1	B2	B3	
Prob	0.6	0.3	0.1	

Alpha	A1	A2	A3	A4
Prob	0.166667	0.666667	0.066667	0.1
Beta	B1	B2	B3	
Prob	0.6	0.3	0.1	

Alpha	A1	A2	A3	A4
Prob	0.166667	0.666667	0.066667	0.1
Beta	B1	B2	B3	
Prob	0.6	0.3	0.1	

Figure 44: Two Different Orderings of Selecting A2 and B2 from Figure 32

The PMF for a matrix with no relationships can be calculated by directly evaluating every combination of conditions. Each combination will have a corresponding probability and effect as defined by Equations 34 and 26 respectively.

Many problems will involve a relationship matrix which affects the probabilities of individual conditions within a matrix. In this case, Equation 37 enforces a normalization of each parameter after the application of relationships. This normalization enforces this rule of probability to ensure that the sum of all potential combinations will sum to unity. However, the fact that probability relationships only propagate to unselected parameters introduces an order dependency to the selection of conditions. This effect can be seen through the selection example in Figure 44. In the left hand column of this figure, A2 is selected triggering a doubling of B2's probability relative to B1 and B3. In the right hand column, B2 is selected triggering a doubling of A2's probability relative to A1, A3, and A4. The selection of A2 and B2 from the left hand column has a probability of 0.23 while the selection of A2 and B2 from the right hand column has a probability of 0.2.

Order dependence results in $n!$ potential ways of selecting a combination where n is equal to the number of parameters in the extended morphological field. This will in turn

result in $n!$ potential probability values for each selection of conditions. Simply selecting each permutation of each combination of conditions will result in a set of selections whose probability values sum to $n!$ because each method of selecting will result in an extended matrix whose total probability will sum to unity. In order to satisfy rule number 2 from Equation 42 and generate a PMF from the selection of each permutation, the probabilities of each selection need to be adjusted to account for the $n!$ ways of choosing a combination from the matrix. The probabilities of each permutation are adjusted using Equation 43; after applying this equation, the summation of the probabilities generated by summary evaluation of all permutations of all combinations will be equal to unity and satisfy rule number 2. A PMF can then be generated using these adjusted probability values and the effect values for each permutation of combinations. The direct calculation of a PMF requires $n! * \prod_{i=1}^n v_i$ evaluations of combinations from the morphological field.

$$P_{new} = \frac{P_{combination}}{n!} \quad (43)$$

5.4.4 Experiment 1– Monte Carlo Simulation of EMA

As cycling through every potential combination of an executable matrix of alternatives is an expensive operation, an alternative is to randomly select a subset of combinations via Monte Carlo simulation– the histogram of the Monte Carlo simulation will resemble the PMF of the fully-sampled executable matrix of alternatives. Hypothesis 3a states that a Monte Carlo simulation can yield appropriate results for less computational cost than a summary evaluation. Hypothesis 3a will be substantiated if a Monte Carlo simulation can adequately reproduce key statistics from the CDF in fewer combination evaluations than a summary evaluation of the matrix.

The key statistics to be tracked in this experiment are the expected value and the 80% quantile. The expected value will be tracked as that is a measure of central tendency which will determine which effect is most likely. The equation for the expected value can be seen in Equation 44. The 95% confidence interval is also calculated in order to track the probability bounds for the estimate; this interval utilizes the square root of the variance or standard

deviation (Equation 45) to calculate the 95% confidence interval (Equation 46). [58, 141]

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (44)$$

$$s^2 = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (45)$$

$$\bar{x} \pm t_{\alpha/2} \frac{s}{\sqrt{n}} \quad (46)$$

The 80% quantile is tracked because it is a recommended value to use when forecasting margins. [2] This value represents a conservative value while not being overly risk averse. This estimate utilizes order statistics for both the quantile estimate and the confidence intervals; the quantile measurement and confidence interval equations can be seen in Equations 47 and 48 respectively. [75, 133]

$$\hat{x}_q = X_{[nq]} \quad (47)$$

$$\begin{aligned} P(X_{(r)} \leq x_q \leq X_{(s)}) &\geq 1 - \alpha \\ r &= [nq - z_{1-\alpha/2} \sqrt{nq(1-q)}] \\ s &= [nq + z_{1-\alpha/2} \sqrt{nq(1-q)}] \end{aligned} \quad (48)$$

5.4.4.1 Monte Carlo Process

The Monte Carlo simulation procedure is visualized in Figure 45. When a new case begins, the first step involves selecting the parameter which from which a condition will be selected; this order must be randomized because the order of selection matters for fields with heterogeneous relationship matrices. Each parameter has an equal chance of being chosen from a random number generated via a uniform distribution (0,1).

The second step is selecting a specific condition from the parameter selected in the previous step. In order to select a parameter, a random number from a uniform distribution (0,1) is generated. This random number is used to select the condition based on the probability

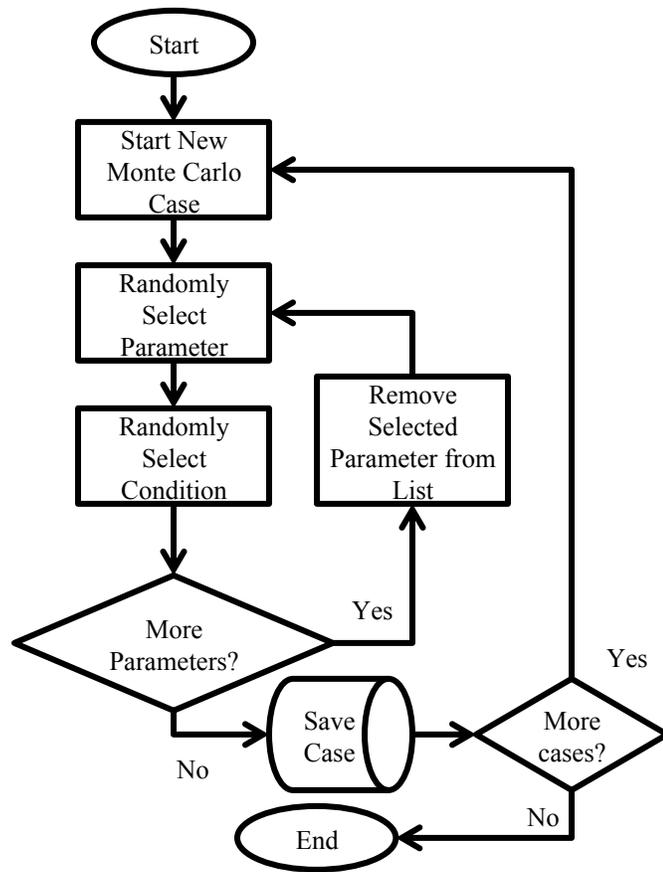


Figure 45: Monte Carlo Procedure for EMA Models

value of the conditions in the parameter. After a condition is selected, the relationships are activated to modify the other parameters of the executable matrix of alternatives.

If there are remaining unselected parameters in the list, then the previously selected parameter is removed from the list of parameters and the parameter selection process starts anew with the updated list. A Monte Carlo simulation case is completed when conditions from all parameters have been selected; after case completion, the selected conditions are saved too the simulation output.

After every completed Monte Carlo case, there is a check to determine if the simulation is complete. This process will continue until the specified number of cases have been completed.

5.4.4.2 Monte Carlo Applied to Example Matrix

The first step in evaluating Hypothesis 3a is making a comparison between the direct calculation of a PMF and CDF and calculation via Monte Carlo simulation. In order to make this comparison, a small morphological field must be chosen to avoid excessive computational complexity. For this test case, the morphological field from Figure 37 and its corresponding relationship matrix in Figure 38 are evaluated.

The first step in this comparison is calculating the CDF for this matrix for the test case where relationships between parameters are omitted. The directly calculated CDF can be seen in Figure 46; this calculation yielded an expected value of 1,286 and an 80% quantile of 1,869.

The CDF for the direct simulation can be compared with the CDF computed via Monte Carlo simulation as seen in Figure 47; this Monte Carlo simulation replicates the CDF calculated via direct-calculation. The Monte Carlo simulations evaluations of these values can be seen for the expected value and 80% quantile in Figures 48 and 49 respectively. In both figures, the value of the statistic if plotted versus the number of Monte Carlo simulation cases used to calculate the statistic; additionally the 95% confidence interval is plotted to bound the expected error in the statistic. The expected value and the 80% quantile values converge after approximately 1,500 and 1,000 Monte Carlo simulation cases

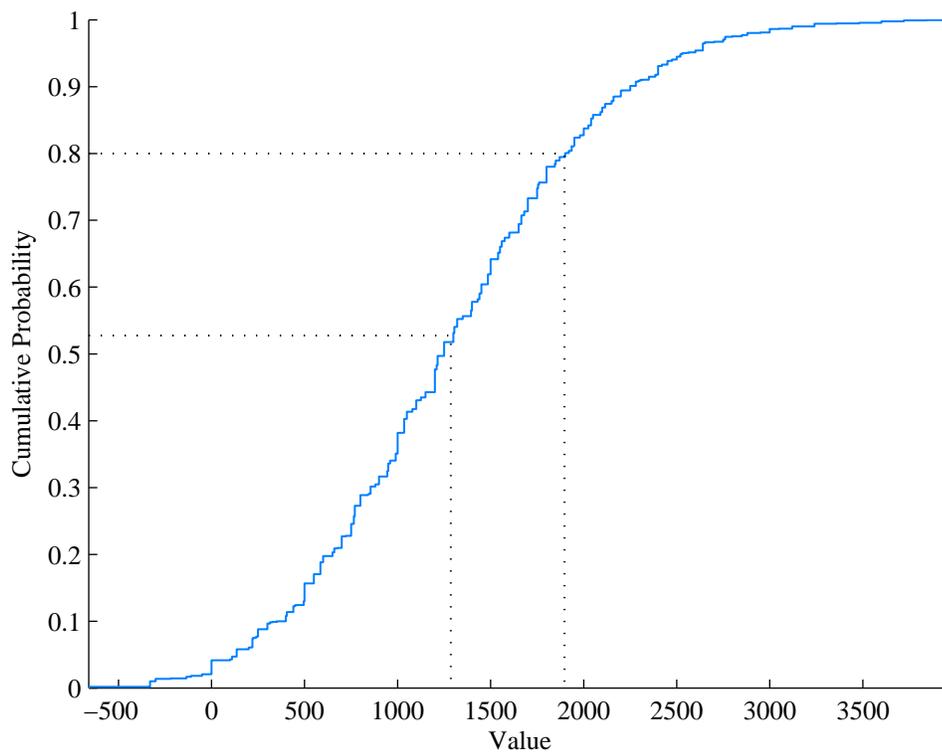


Figure 46: The directly computed CDF for the small example matrix without relationships

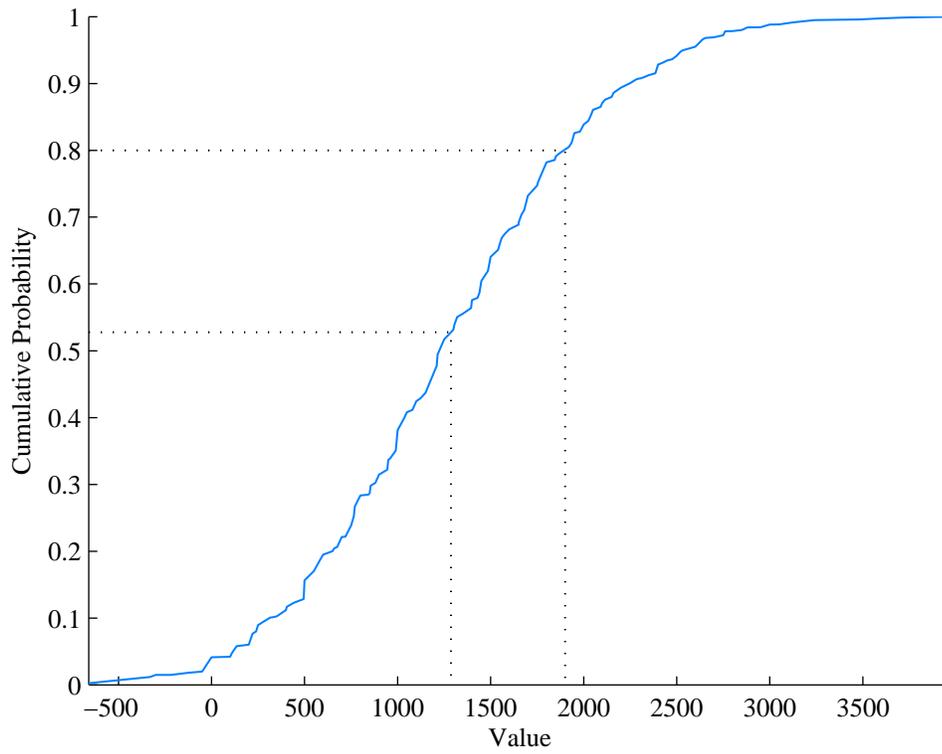


Figure 47: The CDF for the small example matrix without relationships computed via Monte Carlo simulation

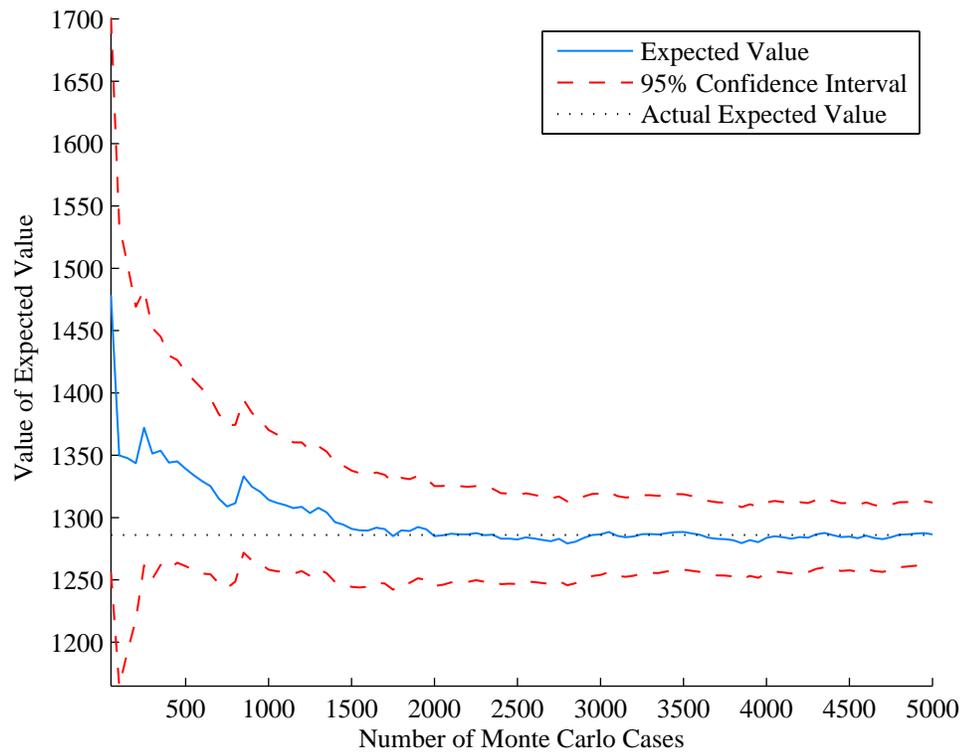


Figure 48: Expected value and confidence interval versus number of Monte Carlo simulations for the small example matrix without relationships

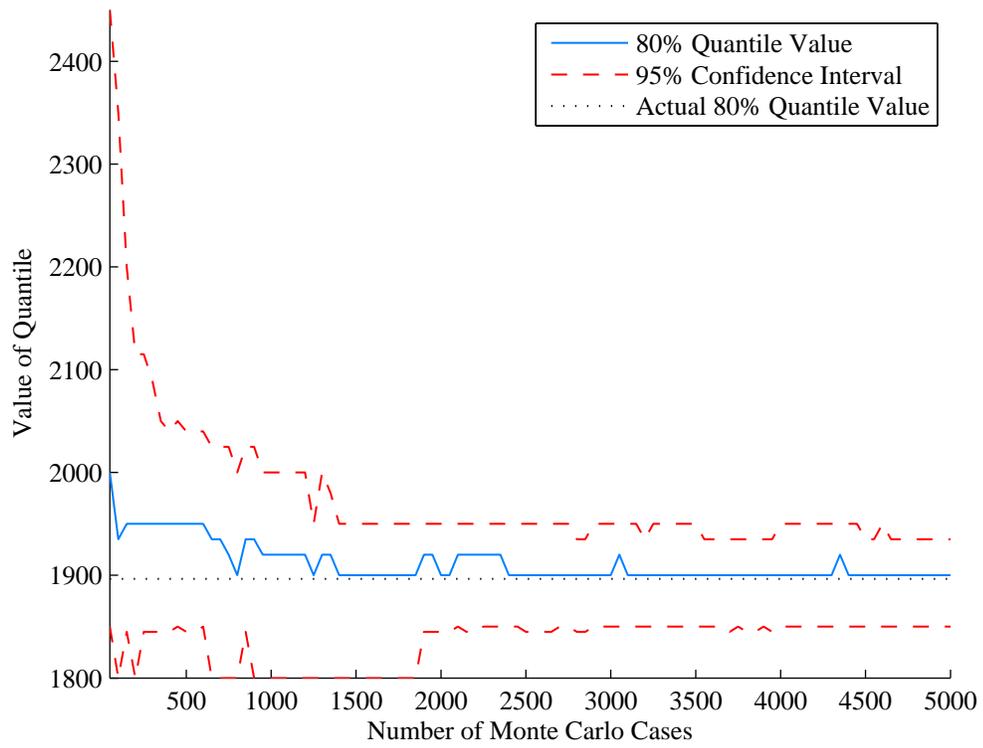


Figure 49: 80% quantile value and confidence interval versus number of Monte Carlo simulations for the small example matrix without relationships

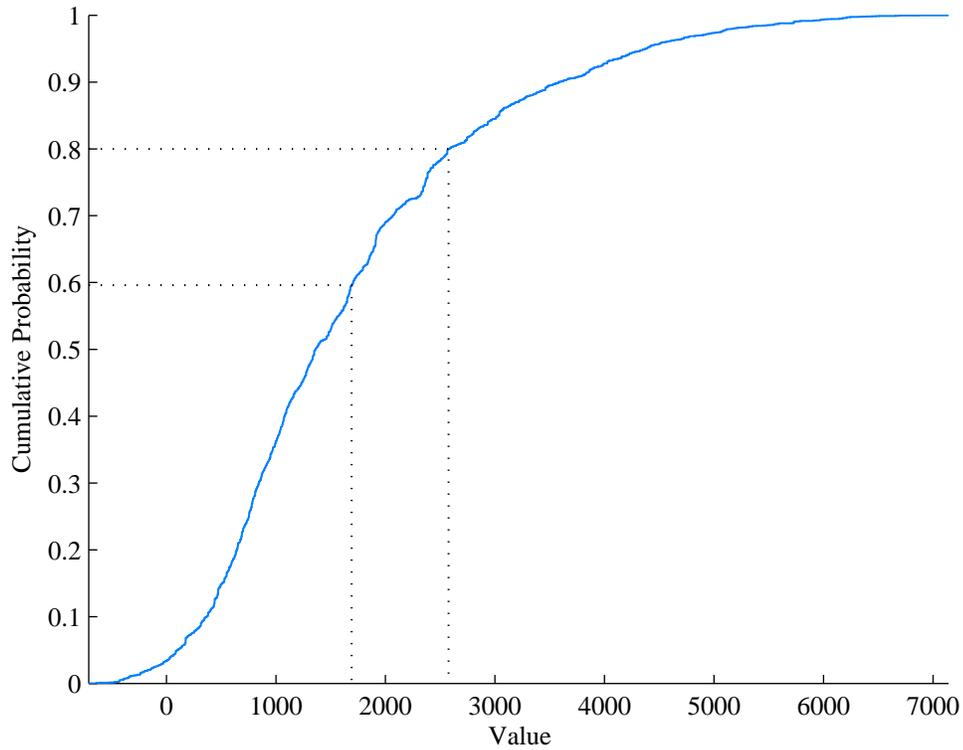


Figure 50: The directly calculated CDF for the example matrix with relationships

respectively. While this is computationally more expensive than a direct calculation, it serves as a benchmark from which the computational complexity required of larger and more complicated matrices can be compared

The next step in this analysis is to add the relationships as depicted in Figure 38. The direct calculation of the CDF will require evaluating every permutation of selections. The directly calculated CDF is shown in Figure 50; this yielded an expected value of 1,690 and an 80% quantile of 2,576. However, in order to generate these results, 43,200 combinations needed to be evaluated.

In comparison to the direct evaluation of every potential combination, a Monte Carlo simulation was used to generate a CDF. The CDF that results from a 5,000 case Monte Carlo simulation can be seen in Figure 51. This CDF has approximately the same granularity as Figure 50 despite requiring only 5,000 evaluations instead of 43,200 evaluations. This calculation Monte Carlo simulation is an order of magnitude less expensive to compute

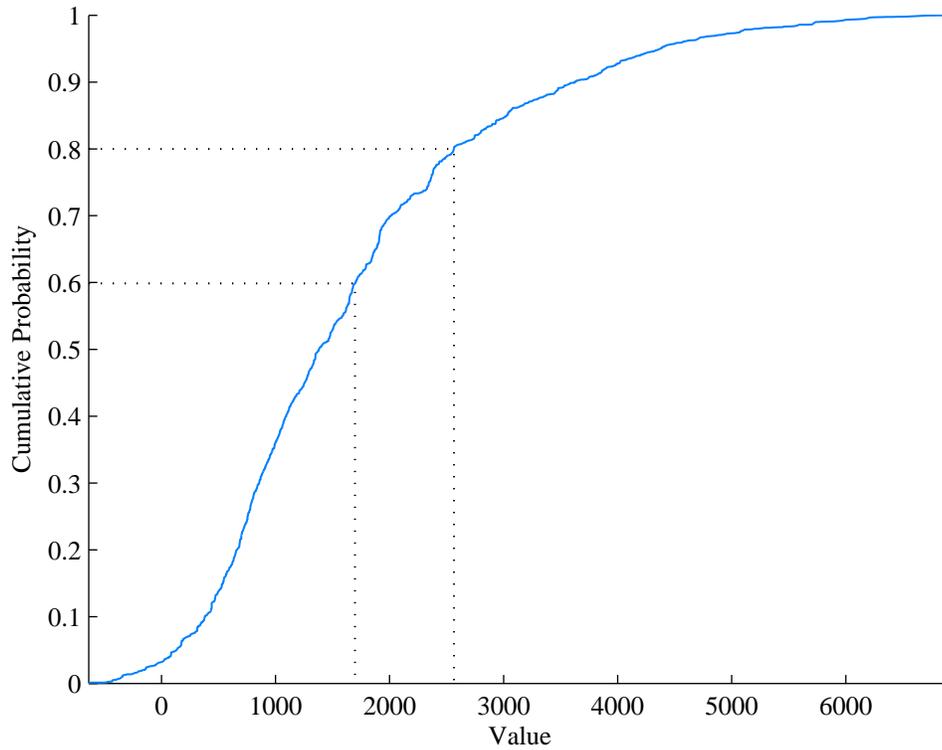


Figure 51: The CDF for the small example matrix with relationships calculated via Monte Carlo simulation

compared to the summary evaluation.

Figures 52 and 53 present the number of Monte Carlo cases versus the key statistics for the case with active relationships. The Monte Carlo simulation converges on the key statistics to within 5% of the direct calculation in approximately 1,500 cases and converges to the actual value in approximately 2,500 even though this matrix has an order of magnitude more combinations. This result further supports Hypothesis 3a in that 2,500 function evaluations is much less than the 43,200 required to fully evaluate the executable matrix of alternatives.

5.4.4.3 Monte Carlo Simulation for Larger Morphological Fields

The evaluation of the Small Matrix has made it clear that a Monte Carlo is less expensive than evaluating the entirety of the executable matrix of alternatives. The next step is to see how the computational expense required to calculate the key statistics scales with the size of

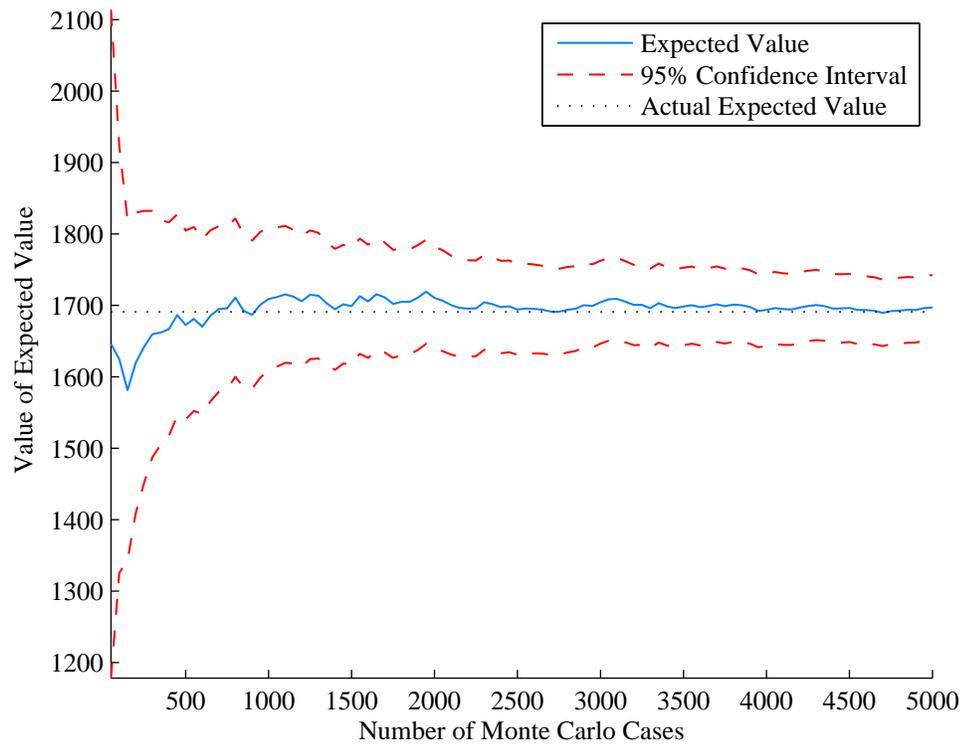


Figure 52: Expected value and confidence interval versus number of Monte Carlo simulations for the small example matrix with relationships

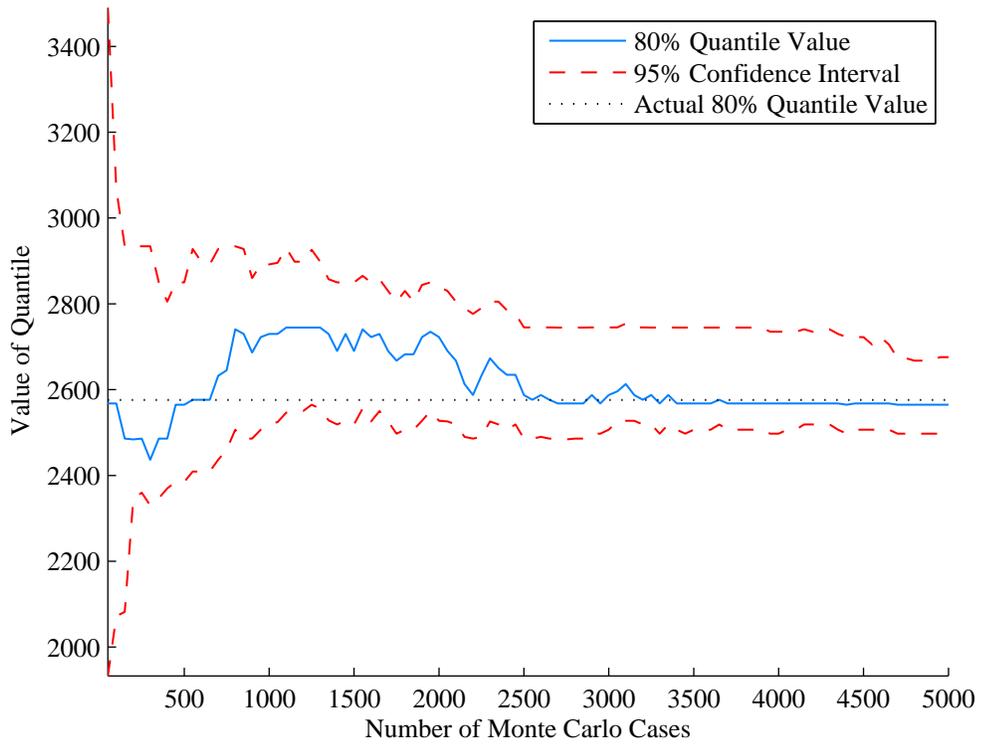


Figure 53: 80% quantile value and confidence interval versus number of Monte Carlo simulations for the small example matrix with relationships

Name	Parameter	A.1	A.2	A.3				
Prob	Alpha	0.2	0.5	0.3				
Name	Parameter	B.1	B.2	B.3	B.4	B.5		
Prob	Beta	0.15	0.15	0.2	0.2	0.3		
Name	Parameter	Γ.1	Γ.2	Γ.3	Γ.4			
Prob	Gamma	0.4	0.4	0.1	0.1			
Name	Parameter	Δ.1	Δ.2					
Prob	Delta	0.6	0.4					
Name	Parameter	E.1	E.2	E.3				
Prob	Epsilon	0.3	0.3	0.4				
Name	Parameter	Z.1	Z.2	Z.3	Z.4			
Prob	Zeta	0.2	0.2	0.25	0.35			
Name	Parameter	H.1	H.2	H.3	H.4	H.5		
Prob	Eta	0.35	0.15	0.1	0.3	0.1		
Name	Parameter	Θ.1	Θ.2	Θ.3				
Prob	Theta	0.4	0.3	0.3				
Name	Parameter	I.1	I.2	I.3	I.4	I.5	I.6	I.7
Prob	Iota	0.1	0.2	0.25	0.1	0.1	0.15	0.1
Name	Parameter	K.1	K.2	K.3	K.4			
Prob	Kappa	0.3	0.4	0.25	0.05			
Name	Parameter	Λ.1	Λ.2					
Prob	Lambda	0.75	0.25					
Name	Parameter	M.1	M.2	M.3	M.4			
Prob	Mu	0.3	0.25	0.25	0.2			

Figure 54: Medium-sized executable matrix of alternatives

the morphological field. This will be determined by conducting Monte Carlo simulations on a medium-sized, 12 morphological parameters, and a large, 24 morphological parameters, executable matrix of alternatives.

The medium-sized executable matrix of alternatives used for this experiment can be seen in Figure 54. This experiment will use pre-defined probabilities, randomly generated additive effects, and randomly generated relationship matrix. The pre-defined probabilities for each condition are shown in Figure 54. The randomly generated relationship matrix has a 80% probability of an additive relationship, a 50% chance of a multiplicative relationship, a 50% chance of an incompatibility, a 50% chance of a probability relationship.

The CDF of the medium-sized executable matrix of alternatives can be seen in Figure

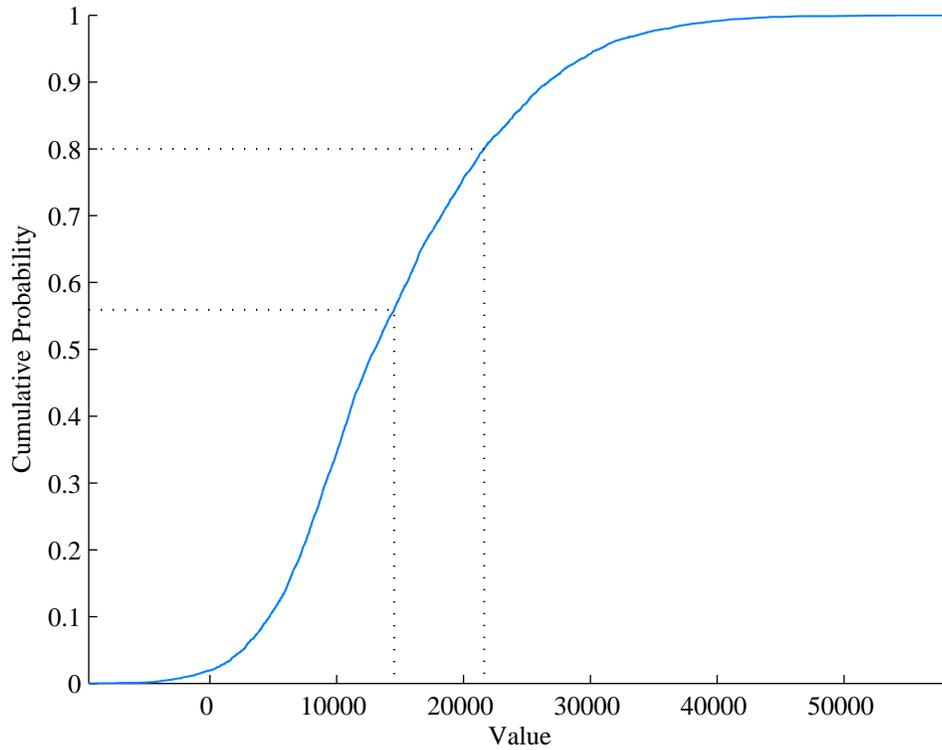


Figure 55: The CDF for a medium-sized morphological field with relationships calculated via Monte Carlo simulation for the medium-sized executable matrix of alternatives

55. This CDF was generated using 5000 Monte Carlo simulation cases; the expected value and 80% quantile values computed after 5000 cases are also highlighted.

Figures 56 and 57 show the plots of the expected value and 80% quantile value versus the number of Monte Carlo simulations respectively. These figures show that the key statistics reach an equilibrium after approximately 1,250-1,500 cases. This result shows that even though this executable matrix of alternatives is much larger, the required number of Monte Carlo cases required is relatively stable. In order to confirm this finding, a 24 parameter morphological field will be run through a similar test.

The 24 parameter morphological field used for this experiment can be seen in Figure 58. As with the medium-sized executable matrix of alternatives, the effects and relationship matrix are randomly generated; the randomly generated relationship matrix has a 80% probability of an additive relationship, a 50% chance of a multiplicative relationship, a 50%

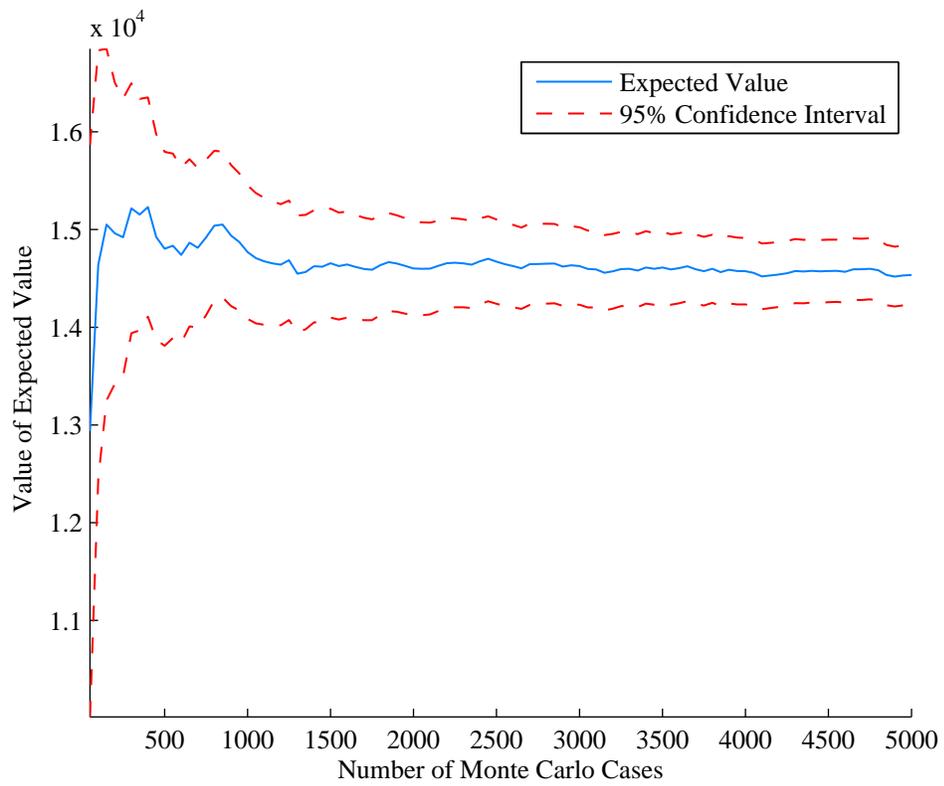


Figure 56: Expected value value and confidence interval versus number of Monte Carlo simulations for the medium-sized executable matrix of alternatives

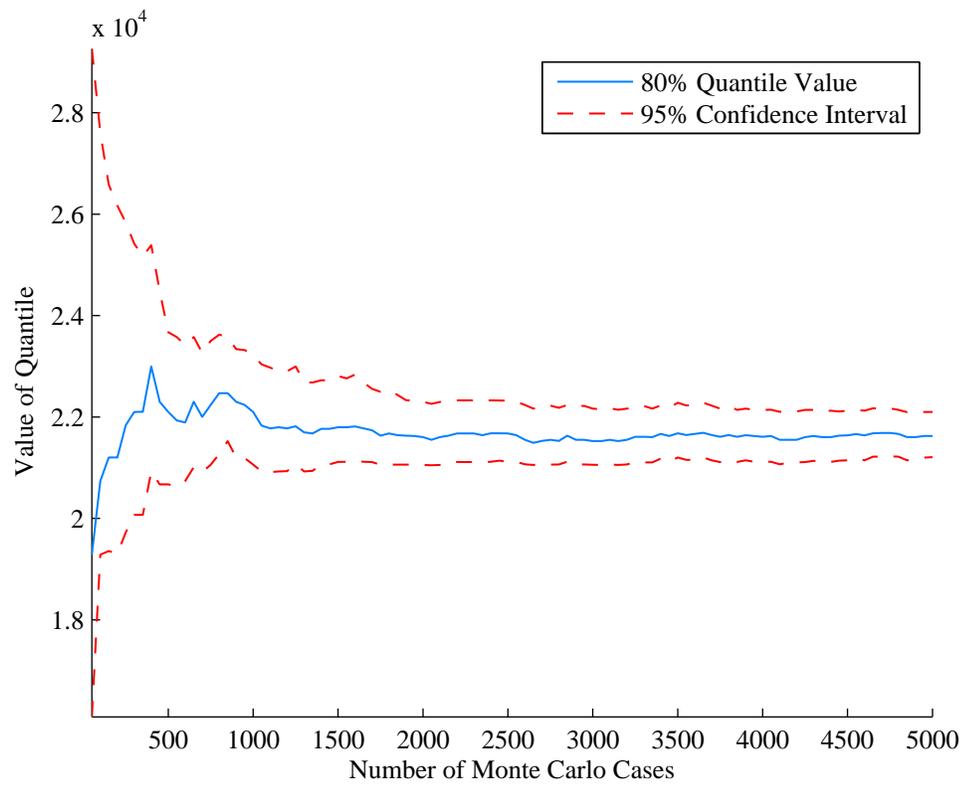


Figure 57: 80% quantile value and confidence interval versus number of Monte Carlo simulations for the medium-sized executable matrix of alternatives

chance of an incompatibility, a 50% chance of a probability relationship.

A Monte Carlo simulation was run over this large matrix for 5,000 cases; the resulting CDF can be seen in Figure 59. The larger number of cases was chosen because it is possible that the larger matrix would require more cases to reach an equilibrium. However, the results of this experiment show that the additional cases were not required. Figures 60 and 61 show the plots of the expected value and the 80% quantile value versus the number of Monte Carlo cases respectively for the large matrix. These figures show that the key statistics tend to converge in 1250-1500 Monte Carlo cases; this result confirms the trend shown in the medium-sized case: the number of Monte Carlo cases necessary to explore an executable matrix of alternatives is insensitive to its size.

5.4.4.4 Conclusions from Experiment 1

The first portion of Experiment 1 presented a direct comparison between calculating all potential combinations of an executable morphological matrix and conducting a Monte Carlo simulation. The results showed that a Monte Carlo simulation required less computational effort than a summary evaluation of a matrix with relationships while maintaining a high level of accuracy. The results of the second part of this experiment further confirm this notion. Larger morphological fields showed an insensitivity to the number of Monte Carlo cases required to generate an accurate CDF. Based on these results, Hypothesis 3a is confirmed: a Monte Carlo simulation is the preferred algorithm to extract probabilistic information from an executable matrix of alternatives. This conclusion comes with a single caveat; for smaller executable matrices of alternatives with homogeneous relationship matrices, a summary evaluation may be computationally inexpensive. When a full computation is inexpensive, it should be used to avoid statistical error.

The most important conclusion which can be reached from this experiment is that useful quantitative information can be extracted from an executable matrix of alternatives without significant computational effort. Whereas previous literature on morphological analysis emphasizes the infeasibility of evaluating different concepts, this result allows EMA to be used for analyzing entire concept spaces. [40]

Name	Parameter	A.1	A.2	A.3				
Prob	Alpha	0.2	0.5	0.3				
Name	Parameter	B.1	B.2	B.3	B.4	B.5		
Prob	Beta	0.15	0.15	0.2	0.2	0.3		
Name	Parameter	Γ.1	Γ.2	Γ.3	Γ.4			
Prob	Gamma	0.4	0.4	0.1	0.1			
Name	Parameter	Δ.1	Δ.2					
Prob	Delta	0.6	0.4					
Name	Parameter	E.1	E.2	E.3				
Prob	Epsilon	0.3	0.3	0.4				
Name	Parameter	Z.1	Z.2	Z.3	Z.4			
Prob	Zeta	0.2	0.2	0.25	0.35			
Name	Parameter	H.1	H.2	H.3	H.4	H.5		
Prob	Eta	0.35	0.15	0.1	0.3	0.1		
Name	Parameter	Θ.1	Θ.2	Θ.3				
Prob	Theta	0.4	0.3	0.3				
Name	Parameter	I.1	I.2	I.3	I.4	I.5	I.6	I.7
Prob	Iota	0.1	0.2	0.25	0.1	0.1	0.15	0.1
Name	Parameter	K.1	K.2	K.3	K.4			
Prob	Kappa	0.3	0.4	0.25	0.05			
Name	Parameter	Λ.1	Λ.2					
Prob	Lambda	0.75	0.25					
Name	Parameter	M.1	M.2	M.3	M.4			
Prob	Mu	0.3	0.25	0.25	0.2			
Name	Parameter	N.1	N.2	N.3				
Prob	Nu	0.1	0.8	0.1				
Name	Parameter	Ξ.1	Ξ.2	Ξ.3	Ξ.4	Ξ.5		
Prob	Xi	0.6	0.1	0.1	0.15	0.05		
Name	Parameter	O.1	O.2	O.3	O.4			
Prob	Omicron	0.3	0.4	0.1	0.2			
Name	Parameter	Π.1	Π.2					
Prob	Pi	0.5	0.5					
Name	Parameter	P.1	P.2	P.3	P.4	P.5	P.6	
Prob	Rho	0.2	0.2	0.3	0.1	0.1	0.1	
Name	Parameter	Σ.1	Σ.2	Σ.3				
Prob	Sigma	0.1	0.1	0.8				
Name	Parameter	T.1	T.2	T.3				
Prob	Tau	0.1	0.4	0.5				
Name	Parameter	Υ.1	Υ.2					
Prob	Upsilon	0.3	0.7					
Name	Parameter	Φ.1	Φ.2	Φ.3	Φ.4			
Prob	Phi	0.2	0.25	0.15	0.4			
Name	Parameter	X.1	X.2	X.3				
Prob	Chi	0.4	0.2	0.4				
Name	Parameter	Ψ.1	Ψ.2					
Prob	Psi	0.4	0.6					
Name	Parameter	Ω.1	Ω.2	Ω.3				
Prob	Omega	0.3	0.5	0.2				

Figure 58: Large executable matrix of alternatives

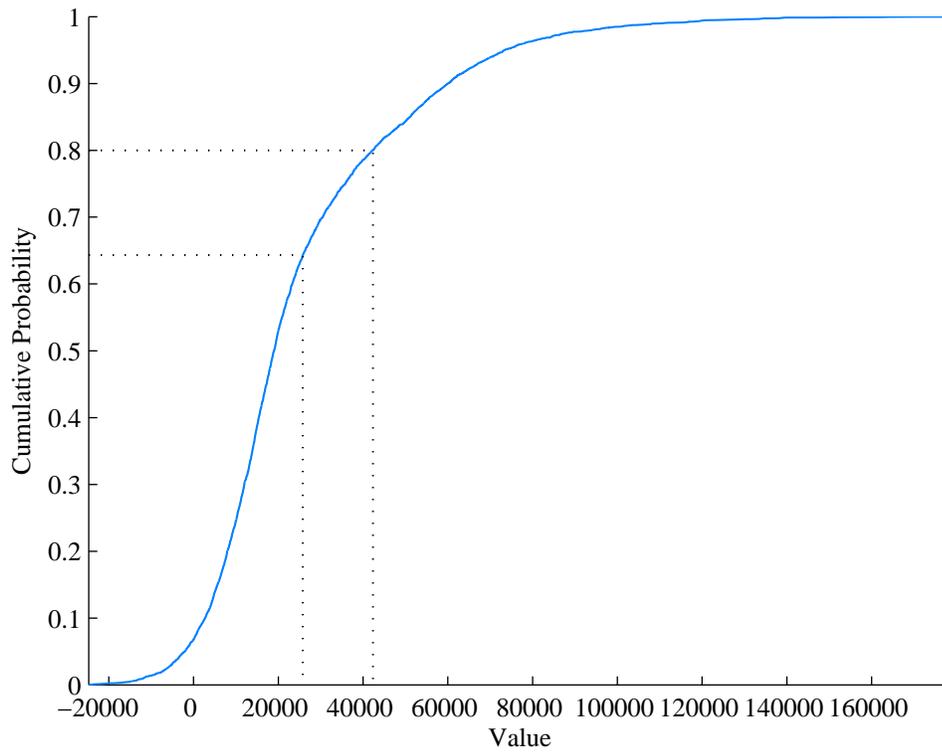


Figure 59: The CDF for a large morphological field relationships calculated via Monte Carlo simulation

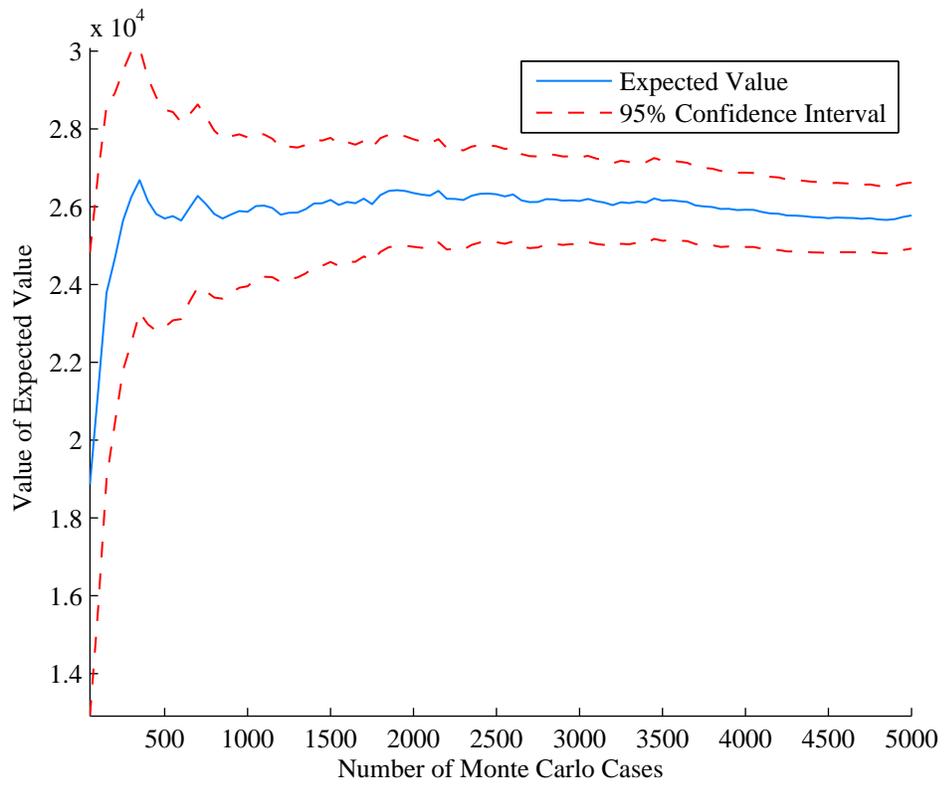


Figure 60: Expected value value and confidence interval versus number of Monte Carlo simulations for the large executable matrix of alternatives

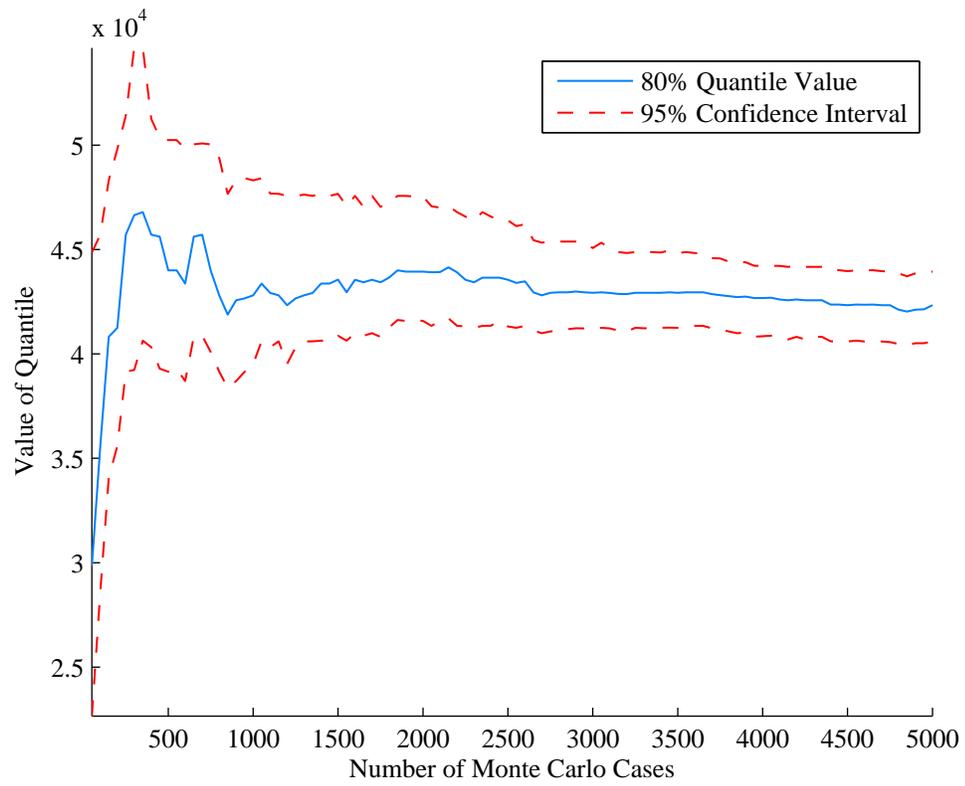


Figure 61: 80% quantile value and confidence interval versus number of Monte Carlo simulations for the large executable matrix of alternatives

5.5 *Implementation of EMA Data Structures*

The previous sections focused on the equations and algorithms which formulate EMA but did not discuss a specific implementation. Previous computerized implementations of traditional MA and extensions of MA have focused on a matrix-based formulation. These implementations are efficient at computing compatible combinations and results based on extensions of the cross-compatibility matrix. [40, 64, 100, 111] However, none of the previous extensions incorporate heterogeneous data, constraints, algorithms, and multifaceted relationships. In order to achieve this extension, an object-oriented approach toward EMA data structures should be used. This is formalized in Hypothesis 3b.

Hypothesis 3b: The inclusion of data, constraints, and complex relationships should be implemented using an object-oriented approach.

This section documents the object-oriented implementation of EMA and show how an object-oriented design acts as an enabler for the extensions required by EMA. The code for this implementation was written in object-oriented MATLAB; the code for the implementations discussed in this section is listed in Appendix A.

5.5.1 **Object Oriented Formulation**

The fundamental principle behind object-oriented (OO) analysis and design is that the data is the primary consideration taken when designing software; the functional behavior of the software is of secondary importance. [104] For EMA, the central pieces of data are the executable matrix of alternatives, the matrix's constituent parameters, the parameters' constituent conditions, and the relationships between conditions.

5.5.1.1 *Generic Implementation*

Because EMA was originally formulated as a generic extension to traditional MA, the OO formulation of EMA begins with a generic implementation from which the specific implementation for a weight forecasting problem can be extended. The generic class diagram can be seen in Figure 62.

The central class in this implementation is the ExecutableMatrix. This class represents the entire executable matrix of alternatives and is responsible for holding a list of

parameters, the relationship matrix, and all methods which operate on the entire matrix. The ExecutableMatrix has seven methods which operate on the class. The delete function is the destructor method. The addParameter and setupRelationships methods are called during the initialization of the matrix; addParameter adds a parameter to the list of parameters, and setupRelationships initializes the relationship matrix. In order to select a single combination, the selectCombination takes in a vector of condition selections; in order to select all combinations, the totalCombinations method would be called while making use of the helper function selectCompatibleCombinations. During the selection of a combination, enforceAllConstraints is used to cycle through each Parameter and call Parameter-level functions which enforce model constraints. Finally, when the selection of a single combination is complete, the resetMatrix method returns both the relationship matrix and all parameters and conditions to their original states.

The Parameter class models a single parameter of an EMA model. This class is primarily responsible for keeping a list of constituent conditions. Additionally, this class has a string which contains the name of the Parameter. This class contains two methods for manipulating the list of conditions. The getCondition method takes in an integer index value and returns the specified condition from the condition list; this method is called by the selectCombination method in the ExecutableMatrix class. The addCondition method adds a condition to the end of the Parameter list; this method is called during the initial definition of the problem. The resetParameter method works is called by the resetMatrix method in ExecutableMatrix in order to reset the state of the model; disconnect also works with the destructor of the ExecutableMatrix class. The most important method in this class is enforceConstraints; this abstract method is responsible for enforcing parameter-level constraints and changing condition properties. This feature of the OO implementation allows a single EMA model to have heterogeneous parameters with different constraints in the same model with minimal code changes.

The Condition class models an individual condition. It includes its name, a list of relationships with other conditions, and a boolean to indicate its modified status. This boolean is necessary for state-dependent relationships. Problem-specific implementations

of a condition will be responsible for adding the data necessary for the problem under consideration. The condition class contains two methods for handling deletion: the delete deconstructor and the disconnect method. The addRelationship method is used by the setupRelationships function during initialization or can be manually called when setting up one-way relationships. The isEqual method returns true if two conditions are equal and false if not equal. The select method calls all of the relationships in the relationship list and activates each relationships. Finally, the resetCondition method is an abstract method which resets the condition to its originally defined state.

The RelationshipMatrix class is a simple class which models a relationship matrix and contains a matrix of Relationship objects. This class is initialized at startup by the setupRelationships method in ExecutableMatrix and has a complicated constructor method. It also contains methods for resetting the relationship states and deconstruction.

The final two classes in the OO implementation are the Relationship and Combination. Because the generic superclasses of ExecutableMatrix and Condition directly use these two final classes, a factory design pattern is used. The factory pattern allows for “creating families of related or dependent objects without specifying their concrete classes.” [45] In this implementation, the RelationshipMatrix creates relationships through a RelationshipFactory, and the ExecutableMatrix creates combinations through the CombinationFactory. At startup, the appropriate problem-specific factories are passed to these classes which then use the interfaces to the generic combination and relationship classes.

The Relationship class, created by a RelationshipFactory, represents a relationship between two conditions. This class knows if it has previously been activated through a boolean; it also knows if the relationship is empty and part of the upper right-hand portion of the relationship matrix. Additionally, it also contains pointers to the two Conditions which are linked by the relationship. Problem-specific attributes will have to be subclassed out for specific implementations. The Relationship class has constructors as required by the RelationshipFactory; it also has deconstructor methods. Most importantly, it has a method, resetRelationship, to reset the state of the relationship, and an abstract method, activateRelationship, which modifies the relationship’s constituent conditions and applies the specific

relationship.

The Combination class does not model a specific part of an EMA model. Instead, it is a class to represent output data from a single selection from the executable matrix of alternatives. Primarily, this class contains a list of Condition objects which represent the selection. The methods in this class enable the creation of a Combination object and the addition of a Condition object. The generic Combination class also provides an isEqual method for comparison purposes and an isFeasible method which check to see if the overall probability of the selected Combination is greater than 0.

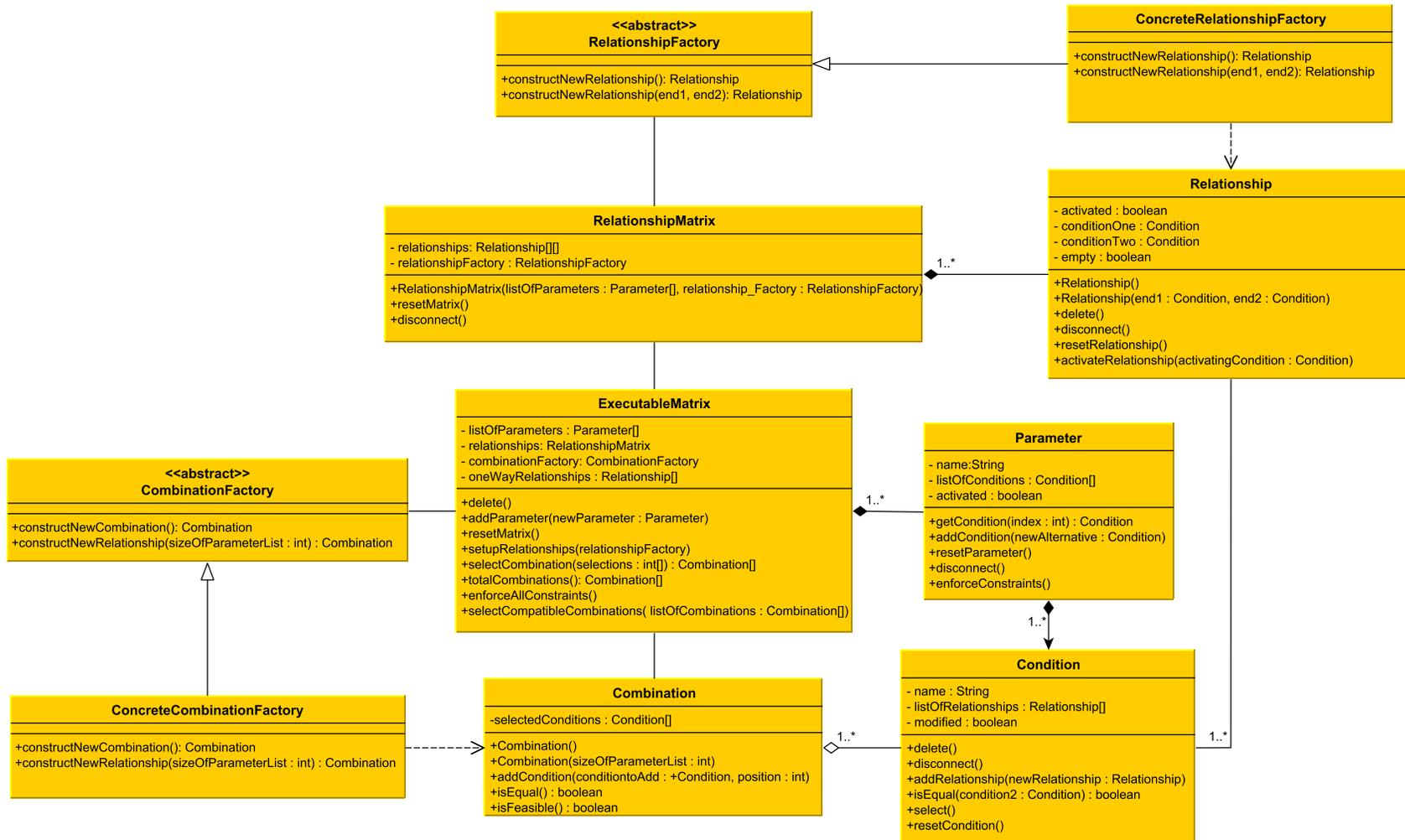


Figure 62: Generic Object-Oriented Implementation

5.5.1.2 *Specific Implementation*

The previous section documented the generic implementation of EMA data structures from which specific implementations can be subclassed. This section documents the extensions to the generic implementation which make up the EMA model used to solve weight forecasting problems. This specific implementation can be seen in Figure 63.

The `ForecastingExecutableMatrix` extends the generic `ExecutableMatrix`. The primary extensions for the weight forecasting problem are methods which extract information from the executable matrix of alternatives. The first method included is `expectedValue`; this method calculates the expected value of an executable matrix of alternatives with a homogeneous requirements matrix through summary evaluation. Similarly, the `generateCDF` function generates a CDF of an executable matrix of alternatives with a homogeneous requirements matrix through summary evaluation. Finally, the `monteCarlo` method conducts a Monte Carlo simulation of the model.

The `ForecastingParameter` class extends the `Parameter` class. This extended class contains a string for identifying the subsystem associated with the parameter; this piece of information is critical if a goal of the forecasting analysis is to analyze weight growth by subsystem. The `subMonteCarlo` works with the `monteCarlo` method in `ForecastingExecutableMatrix` to run a Monte Carlo simulation; this method selects an individual condition from the parameter for selection. Finally, the `enforceConstraints` method overrides the superclass's method in order to implement Equation 37.

The `ForecastingCondition` class extends the `Condition` class. This extension includes the `probability`, `additiveEffect`, and `multiplicativeEffect` attributes for the implementation of this problem-specific EMA. There are three other attributes, `modifiedProbability`, `modifiedAdditiveEffect`, and `modifiedMultiplicativeEffect`, provide a mechanism for the attributes to change during model execution. At the end of an EMA model execution, the modified attributes represent the final state of the attributes and are used to calculate overall probabilities and effects. The `resetCondition` method is called when the executable matrix of alternatives needs to be reset to the initial, as defined state; this method simply copies the original, assigned probability and effects to the attributes labeled as 'modified' and resets

the modified boolean. Finally, the `isEqual` method provides an updated method to establish equality; specifically, it compares the original, unmodified attributes.

The `ForecastingRelationship` class extends the generic `Relationship` class and is produced in a `ForecastingRelationshipFactory`. The `ForecastingRelationship` includes attributes to implement the five relationships described in Section 5.3.2. Constructors are implemented for use with the factory pattern. Finally, the `activateRelationship` method is overridden to implement the equations described in Section 5.3.2.

The final extension creates the `ForecastingCombination` class to extend the `Combination` class. This class is created by the `ForecastingCombinationFactory` in accordance with the factory pattern. The `addCondition` method is overridden in this subclass adds the input `Condition` object to the combination by making a copy of the contents of the original object. Additionally, the `ForecastingCombination` subclass adds methods for calculating the total probability of the selected combination and the total effect of the combination.

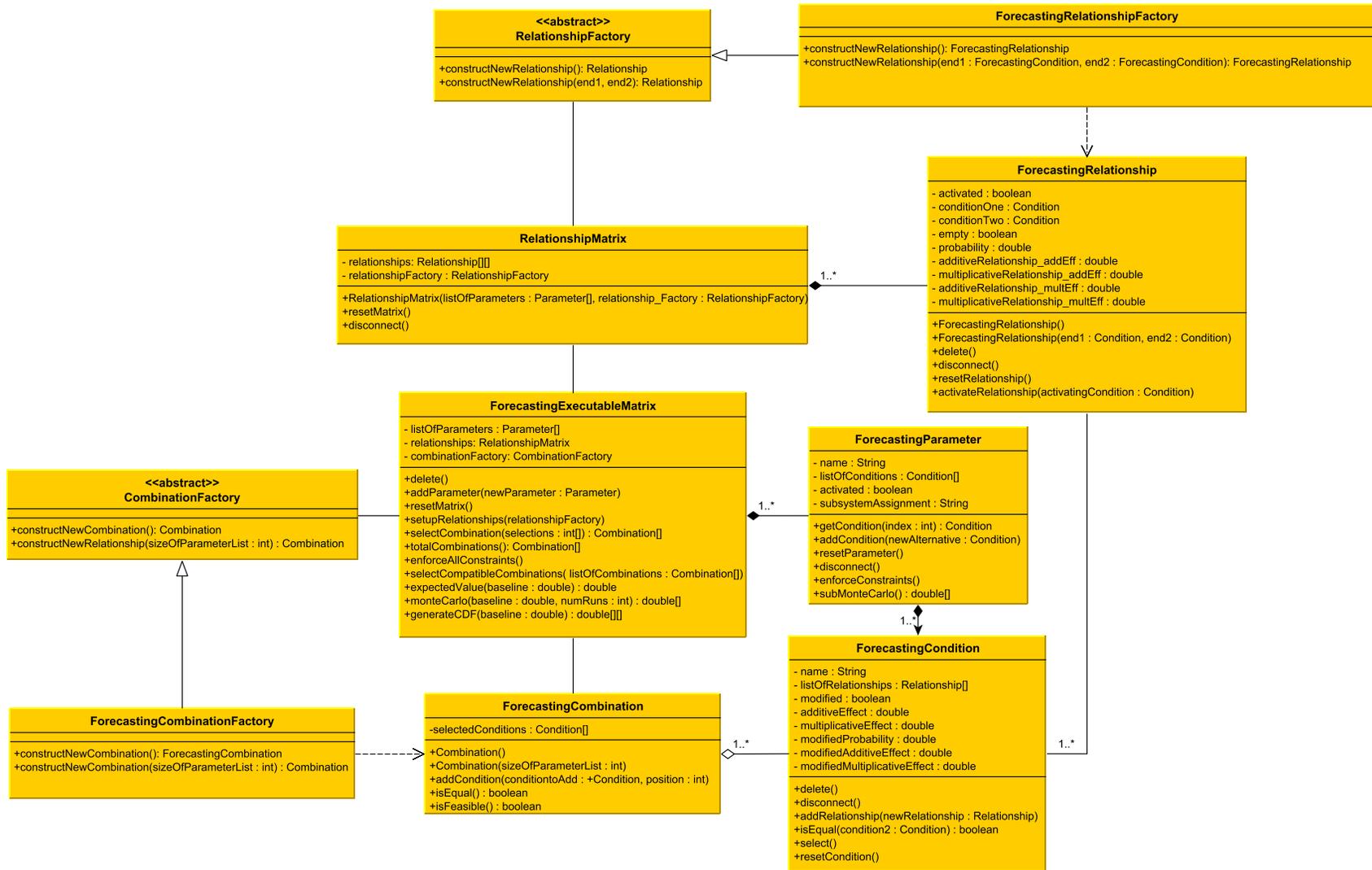


Figure 63: Specific Object-Oriented Implementation for a Weight Forecasting Problem

5.5.2 Running EMA

5.5.2.1 Problem Setup

The first step in defining an EMA forecasting problem is to define the conditions in the model. For forecasting problems, `ForecastingCondition` is the subclass of `Condition` that is used. Each `ForecastingCondition` must be initialized in a setup script. For each defined condition, the name, additive effect, and multiplicative effect should be defined. All other attributes in this class are used during the execution of the model.

Once the conditions of the model have been defined, they need to be added to parameters. First, parameters need to be created in the setup script. Each condition should be added to the parameter via the `addCondition` method in the `ForecastingParameter` class. In addition to adding constituent conditions, the name of each parameter may also be defined. Furthermore, if the problem wants to track subsystem weights, the subsystem assignment needs to be defined in each `ForecastingParameter`.

Once each parameter is defined and filled with constituent conditions, the parameters of the model need to be added to the `ForecastingExecutableMatrix`. This begins with defining a `ForecastingExecutableMatrix` in the setup script. Next, each parameter should be added to the `ForecastingExecutableMatrix` through the use of the `addParameter` method. Once the parameters have been added to the `ForecastingExecutableMatrix`, the factories for combinations and relationships need to be created. For this problem, a `ForecastingCombinationFactory` and `ForecastingRelationshipFactory` are defined. The `ForecastingCombinationFactory` is added to the `ForecastingExecutableMatrix`. Finally, the `setupRelationships` method is called, and the pointer to the `ForecastingRelationshipFactory` is passed to the method; this method sets up the relationship matrix, defines relationships, and assigns relationships to each condition.

The `setupRelationships` method defines all of the relationships present in the relationship matrix to have default 'no effect' values. The next step in the definition of an EMA forecasting problem is to assign relevant values to the attributes of the previously created relationships. This will involve specifying probability, multiplicative, and additive relationships which affect the attributes in `ForecastingConditions`.

Finally, any one-way relationships which are present in the model should be created. This involves defining a `ForecastingRelationship` independently of the relationship matrix. `ConditionOne` in a `ForecastingRelationship` should point to the condition to be selected, and `ConditionTwo` should point to the adjoining condition. The empty boolean should be set to false. Once these attributes have been set, the relevant relationship values can be set to the new relationships. Once a relationship is created, it should be added to the list of `oneWayRelationships` in the `ForecastingExecutable` matrix.

5.5.2.2 Execution of EMA Model

In order to generate quantitative results from the EMA model, the `ForecastingExecutableMatrix` class provides methods which will execute the EMA model. This primarily consists of three methods: `monteCarlo`, `generateCDF`, and `expectedValue`. Each method accepts the *Weight_{remaining}* value from Equation 26. The `expectedValue` method outputs the `expectedValue` of the matrix by summarily selecting all potential permutations of combinations, calculating the effect of each selection, and using Equation 20 to calculate the expected value. In order to generate the probabilistic data necessary to calculate quantiles and display a CDF, the `monteCarlo` and `generateCDF` methods are needed. The `generateCDF` method directly calculates a CDF through summary evaluation of all permutations of combinations; once the effect and probability have been calculated for every potential option, this method then sorts this information to output a table of data containing the CDF. Finally, the `monteCarlo` method executed a Monte Carlo simulation over the data structure as described in Section 5.4.4.1; the output of this method is a table containing the output values and the selections of each Monte Carlo case.

While `monteCarlo` and `generateCDF` will output tables of probabilistic data. If only a single combination is desired, the `selectCombination` method accepts an array of integers corresponding to the desired conditions in each parameter and outputs an array of the permutations of the desired selection. This method can be used as a standalone method if a user wishes to see the impacts of a single combination. Additionally, this is used as a helper method for `generateCDF` and `expectedValue`.

Each of these methods relies on selecting a condition. The selection process starts by calling the `select` method in the desired condition object; this method in turn iterates through all of the relationships involving the selected condition and activates each relationship. The activation of each relationship uses the `activateRelationship` method in the `ForecastingRelationship` class; this method applies the relationship values and updates the state of both the relationship and condition to reflect the activated condition. Updated condition attributes are assigned to the modified attributes in the `ForecastingCondition`. Once the relationships have been applied, the `enforceAllConstraints` method is called in the `ForecastingExecutableMatrix`. This method cycles through each parameter and calls the `enforceConstraints` method which applies Equation 37. This process is repeated for each subsequent condition selection until a condition has been selected from each parameter. In the course of executing a Monte Carlo simulation or a summary evaluation, the matrix will need to be reset; the `resetMatrix` method in `ForecastingExecutableMatrix` cycles through each object in the data structure and restores it to its original state.

5.5.3 Specific Enablers of EMA

As formulated in Sections 5.3 and 5.4, the implementation of EMA requires a much more capable approach than previous matrix-based implementations. The OO implementation used in this thesis addresses the requirements of EMA and provides an elegant and capable implementation of EMA data structures and algorithms.

The most fundamental requirement of EMA is the ability of conditions to hold additional pieces of information other than its name. While this requirement could be satisfied via additional matrices or structures, the OO implementation is more elegant because it stores data at a level where it can be accessed by appropriate methods. For example, the `activateRelationship` method in `ForecastingRelationship` makes use of its internal activated boolean state, the relationship attributes stored in the `ForecastingRelationship` class, and the attributes stored in both `ForecastingCondition` classes. This implementation also allows for parts of the model to have attributes which would normally be overlooked in previous implementations; for example, each parameter can know the subclass to which it is assigned

for later post-processing.

The key feature of EMA enabled by the OO design is the relationship system. In this system, each condition knows its relationships with other conditions, and each relationship knows which conditions it links. This also allows for one-way relationships which exist outside of the relationship matrix. In this system, relationships are activated by a condition object; the act of selecting a condition triggers the condition object to cycle through its list of relationships activating each one. When a relationship is activated, the relationship object can detect the calling condition and both conditions' states; this information can then be used to calculate modifications to each condition. This results in far more elegant algorithms compared to trying to perform the same operations utilizing a large number of different matrices.

Another strength of this implementation is the ability to generalize it to other problems. The generic EMA implementation was specifically designed to be subclassed out in order to handle a wide variety of problems; the forecasting of weight is just one potential application for this concept.

The specific feature of EMA which combines all of the above features of OO design is the constraint system. Each parameter is allowed to enforce constraints on its constituent conditions. This requires a method in the parameter class which has access to information contained in all of its constituent conditions and the ability to modify the states of said conditions. The implementation of this system is enabled by the OO design. Furthermore, this design would allow for a single executable matrix of alternatives to have heterogeneous parameters with different constraints because the ExecutableMatrix class only operates on abstract parameters.

All of these examples show that EMA's implementation is enabled by an OO design. Therefore, Hypothesis 3b is substantiated— the OO implementation is an advancement over previous extensions of traditional MA.

CHAPTER VI

METHODOLOGY DEVELOPMENT

The data structure and algorithms defined in Chapter 5 provide the analytical tools necessary to analyze potential configuration changes. However, to maximize effectiveness, these tools must be used as part of a larger, structured methodology. This structured methodology takes the form of a hybrid methodology as hypothesized by Hypothesis 1 and combines range estimating and EMA. This hybrid methodology is designed to be used during the later phases of conceptual design.

This chapter develops the steps necessary to conduct the hybrid methodology combining range estimating and EMA. The complete hybrid methodology is shown in Figure 6. As range estimating is a well-developed method (developed in Section 3.5.1.2), the focus of this chapter will be on the development of EMA as part of the hybrid methodology. The first section of this chapter describes the inputs necessary for both range estimating and EMA. The second section describes the steps necessary to create and analyze an EMA model. The final section of this chapter describes how to use the output of EMA and range estimating to calculate desired weight and margin values.

6.1 Problem Setup

The hybrid methodology begins with setting up the forecasting problem. The first thing that needs to be established during the problem setup is vehicle's WBS and baseline weight breakdown. This WBS will be used in both range estimating and EMA. In range estimating, the WBS will serve as the basis for the range estimating analysis. In EMA, the WBS will inform the identification of model parameters and condition attributes. In this hybrid methodology, the WBS should serve as a unifying document behind both range estimating and EMA—the subsystem and total weight forecasts from both parts of this hybrid methodology should be directly tractable to the original WBS in order to enable more effective program risk management.

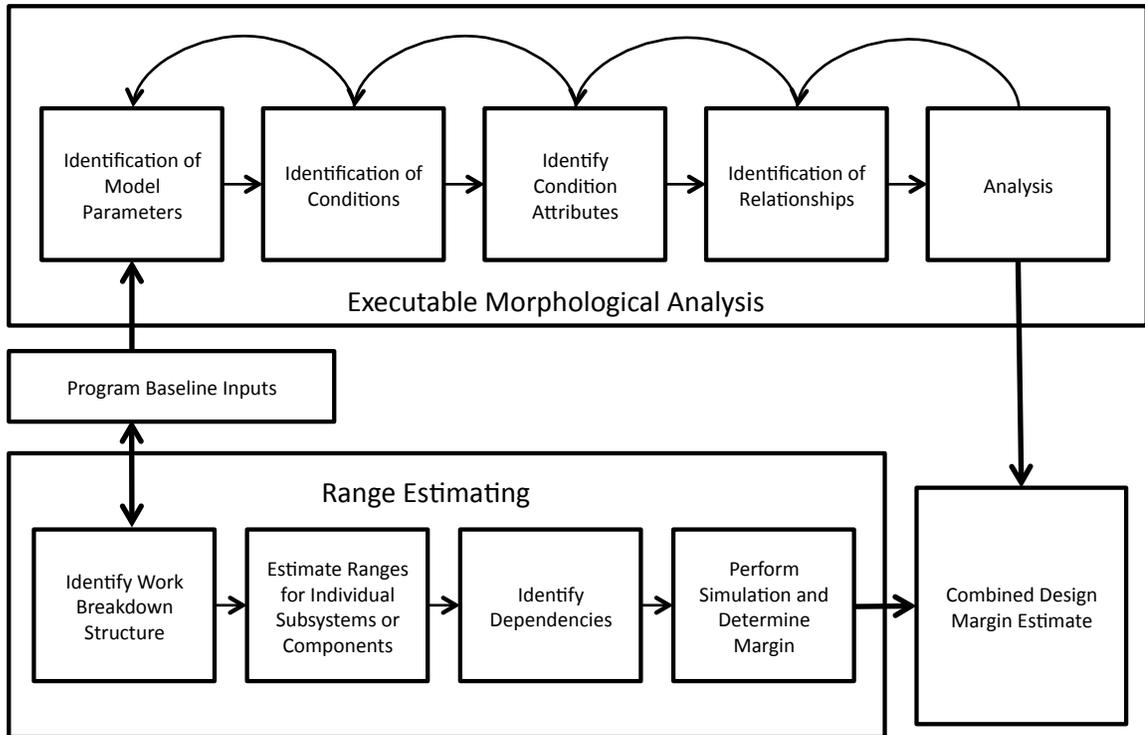


Figure 64: Hybrid Forecasting Method

While range estimating primarily requires the vehicle baseline WBS, EMA requires more information from the vehicle development team. At its core, EMA is a structured way of managing subsystem alternatives for the purposes of risk management. During the development of the vehicle baseline, both the vehicle and subsystem offices conduct trade studies examining subsystem alternatives; this information can be leveraged during all phases of the creation of an EMA model.

In addition to studies into subsystem alternatives, EMA requires knowledge about the assumptions which led to the baseline. This can take the form of the original requirements, technology assumptions, and funding profile. These assumptions describe specific uncertainties which can affect the vehicle program— requirements, technology, and political uncertainty respectively. Because these assumptions can change during the course of the development program, a vehicle program office will have likely performed risk impact and mitigation studies. The output of these studies can be incorporated into the EMA model to allow for unified risk and alternative management.

6.2 Forecasting Utilizing Executable Morphological Analysis

EMA has five steps as outlined in Figure 6: the identification of model parameters, the identification of conditions, the identification of condition attributes, the identification of relationships, and analysis. Figure 6 shows that all of these steps are iterative. While this section focuses on a linear progression of the method, in practice, previous steps should be revisited to ensure the most complete model.

6.2.1 The Identification of Parameters and Conditions

The first two steps of forecasting utilizing executable morphological analysis mirror the first phase of creating a traditional morphological model, the analysis phase; this phase involves the development of the morphological field. According to Ritchey, this is the most important phase of creation of morphological models because this phase shapes the potential problem space. The analysis phase begins by identifying the important dimensions of the problem; these dimensions become the parameters of the morphological field. Once the parameters have been established, the next step in this phase is to assign alternatives to each dimension: the identification of conditions. [110, 111] This section discusses the specifics of creating a traditional morphological model for the purposes of incorporating it into an EMA model for mass forecasting and risk mitigation.

6.2.1.1 Identification of Model Parameters

The first step of building an EMA model is the identification of the forecasting model's parameters. The goal of this phase is to define the parameters which will account for all forms of uncertainty. Because range estimating accounts for some forms of model uncertainty, the focus of EMA should be on technological, political, requirements, volitional, phenomenological, integration, and any model uncertainties which result in baseline changes (see Section 2.3 for a description of these uncertainties).

The first step in defining the parameters for an EMA model is to perform a physical decomposition of the vehicle. A physical decomposition involves decomposing the vehicle into sub-assemblies and components; this decomposition should allow a one-to-one mapping

between range estimating sub-assemblies and EMA parameters. In addition to subsystems identified by the WBS, any subsystem or component not included on the existing on the WBS but identified as a possible necessity based on risk mitigation studies should be included. These additional parameters will enable the EMA model to account for systems which would be unaccounted for by range estimating.

When a series of parameters which can describe the underlying vehicle has been established, the immediate next step is to identify categories of scenarios which will affect the vehicle development program. These categories should describe things which fall outside the direct control of the traditional vehicle development office; these can include changes in requirements, funding profile, schedule, etc. These scenarios can also be used to model technology development programs necessary for the vehicle.

A final step in the identification of model parameters is optional. If the goal of the study is to assign margins to specific subsystems, then each parameter should be labeled with its parent subsystem. If a parameter cannot be assigned to a single subsystem, then it should be assigned to a category representing the entire vehicle; for example direct increases in mass due to budget cuts cannot be assigned to a subsystem and should be attributed to the entire vehicle. These labels will be used during the Monte Carlo simulation to generate subsystem-specific CDFs.

6.2.1.2 Identification of Conditions

The next step in EMA is a continuation of the analysis phase from traditional MA. Now that the model parameters have been defined, the conditions for each parameter need to be defined. The first sub-step in this process is to define the current baseline for all of the parameters identified in the previous step of EMA. This will serve as a basis from which additional configurations can be identified.

The next sub-step involves identifying conditions for the parameters which describe the external influences on the vehicle development program. This sub-step can leverage any risk mitigation studies conducted by the vehicle development office. It may also require expert elicitation from customer or government representatives to describe potential alternative

requirements. The completion of this sub-step will result in a morphological field which models discrete external influences that can reasonably influence the program.

The vehicle analysis portion should start by thinking of alternative development scenarios based on the current design fidelity level— which portions of the design may require redesign due to increased model fidelity? This will involve examining technology assumptions, design decisions, and analyses concerning the more extreme portions of the vehicle’s design envelope. This re-examination of the design should set the stage for brainstorming of alternative conditions which may be necessary to overcome shortcomings in model fidelity. These alternative conditions also need to manifest the result of technology development programs.

The analysis of vehicle alternatives continues by leveraging previous trade studies conducted by the vehicle office and subsystem offices. Alternatives previously ruled-out by trade studies should be included as conditions in EMA if there is a reasonable chance of revisiting previous trade studies. As with the previous sub-step, the conditions which describe differing funding levels and customer priorities should be reflected in these analyses. Changes in requirements can directly lead to different design decisions— any alternate design decisions motivated by different requirements should be explicitly modeled.

Because the sub-steps in this portion of EMA are interdependent, iteration will be necessary. The end result of this step is a morphological field which describes both the development program and vehicle. Furthermore, the alternative scenarios describing the overall vehicle development program (budget, requirements, technologies, etc.) should have vehicle-level conditions which will directly result from changes to the development program. These conditions will be used explicitly during the identification of relationships.

6.2.2 Identification of Condition Attributes

Once the analysis phase of traditional MA has been completed, the next step is EMA specific: identifying the attributes of conditions. For each condition, the probability, multiplicative effect, and additive effect will have to be identified.

The first attribute which must be assigned for each condition is its probability value. As

described in Section 5.3.1.2, the probability of a given condition must be specified in concert with the other conditions in its constituent parameter— the probability values across the parameter should reflect the relative chances of occurrence and sum to unity. Furthermore, the assignment of probability is subjective, therefore expert elicitation should be used to determine the ratios of probabilities between conditions in a parameter.

The next attribute which should be assigned to each condition is the additive effect. This effect describes a scalar addition or subtraction to a baseline weight. In practice, this value can refer to a difference in weight relative to a baseline weight or a subsystem weight value. If a delta-weight is used, then the value of the additive effect in the condition corresponding to the current vehicle baseline would equal 0. If a subsystem weight value is used, then the value of the additive effect in the condition corresponding to the current vehicle baseline would equal the weight allocation in the WBS. These decisions must be documented in order to generate the $Weight_{remaining}$ input to Equation 26; this input is calculated according to Equation 25 where the $Weight_{remaining}$ input to Equation 26 is equal to the WBS total weight minus the sum of additive effects in conditions corresponding to the WBS baseline.

$$Weight_{remaining} = Weight_{baseline} - \sum_{i=1}^n E_{additive_{baseline_i}} \quad (25)$$

$$E_{total} = (Weight_{remaining} + E_{additive_{combination}}) * E_{multiplicative_{combination}} \quad (26)$$

The third attribute necessary for this analysis is a multiplicative effect. As defined by Equation 26, this acts as a scalar multiplication on the weight of the entire vehicle. Because this can represent a dramatic change in mass across the entire vehicle, comparatively few conditions will have an active multiplicative effect. Examples of conditions which may express a multiplicative effect include alternative performance requirements or the implementation of a weight control program.

Unlike the identification of probability attributes, the additive and multiplicative effects can be traced directly to previous design trade studies and risk mitigation efforts carried

out by a vehicle or subsystem office. The results of these studies can be used to inform these attributes and save the information created during earlier phases of design. For most problems, there will be options examined during the creation of an EMA model which have not previously been examined. In order to determine these attributes, a risk manager can assign a 'tiger team' to perform a quick study and generate a high quality result. If the program is lacking on budget or time for additional studies, expert elicitation can be used to determine these attributes.

Many conditions will have no first order effect on the weight of the vehicle. These conditions will have second order effects which will affect the weight of the vehicle through relationships. These relationships are addressed in the next step of the methodology.

6.2.3 Identification of Relationships

The next step in the development of an executable matrix of alternatives extends the synthesis phase of traditional morphological analysis. In the synthesis phase, the total number of simple configurations, the "problem space," is trimmed down to a smaller "solution space" of compatible combinations. In order to reduce the original problem space, a cross consistency assessment is performed on the original morphological field. This assessment consists of pair-wise compatibility assessments between individual conditions. As described in Section 4.2.2.1, this process is enabled by the quadratic growth of pair-wise combinations compared to the geometric growth of simple configurations. [110, 111]

While the synthesis phase in traditional MA only dealt with binary compatibility relationships, the identification of relationships phase in EMA must identify continuous compatibilities and relationships affecting multiplicative and additive effects. While it is possible to have a model with no relationships, only probability relationships, only effectual relationships, or any combination of the above, the most powerful models will have all three types of relationships.

The identification of relationships has three distinct substeps. First, the parameters must be broken into sub-matrices of alternatives; this is meant to organize the problem in such a way as to minimize further analysis. Once a set of matrices has been determined,

one-way relationships can be mapped between conditions of separate matrices. Finally, the two-way relationships within each sub-matrix can be identified.

6.2.3.1 Organization of Parameters

For the specific problem of trying to investigate potential baseline configuration changes, a large and diverse set of scenarios exists. Many of these scenarios, such as changes to the political environment surrounding a development program, do not have a first order effect on potential baseline configuration changes. However, such scenarios do have a second order affect on the development of a vehicle. In EMA, these second-order effects are modeled through one-way relationships. While one-way relationships are necessary to address all relevant forms of uncertainty, they make analyses much more complicated because they dramatically increase the effort necessary to define relationships between conditions.

Minimizing one-way relationships is a necessary step in the setup of a forecasting problem that utilizes EMA. For this problem, the observation can be made that there are two different types of scenarios which can play out: scenarios which affect the environment surrounding the vehicle development program, and scenarios which affect the development of certain vehicle subsystems. Scenarios affecting the conditions surrounding the program would include events such as budget cuts, requirements changes, or other external influences. Scenarios which affect the development of vehicle subsystems would primarily concern the implementation of specific subsystems, the implementation of a mass-saving technology, or a design change due to higher-fidelity analyses.

Given these two categories of scenario elements, another observation can be made: two-way relationships will likely not exist between the two categories, and one-way relationships from subsystem development to higher-level scenarios will be nearly non-existent. The majority of relationships between the two categories will be one-way relationships of program-level scenarios which affect vehicle-level scenario elements. Based on these observations, the assumption can be made that vehicle-level scenarios will not affect program-level scenarios. This assumption allows the forecasting problem to be set up in a way to minimize one-way relationships because the only one-way relationships would be from program-level elements

Name	Parameter One	1.1	1.2	1.3	1.4
Probability		0.35	0.15	0.25	0.1
Effect		0	0	0	0
Name	Parameter Two	2.1	2.2	2.3	
Probability		0.35	0.15	0.25	
Effect		0	0	0	

Name	Parameter Alpha	A.1	A.2	A.3		
Probability		0.25	0.25	0.5		
Effect		400	-200	0		
Name	Parameter Beta	B.1	B.2	B.3	B.4	B.5
Probability		0.35	0.15	0.25	0.1	0.15
Effect		0	500	250	-100	350
Name	Parameter Gamma	Γ.1	Γ.2	Γ.3	Γ.4	
Probability		0.25	0.25	0.25	0.25	
Effect		200	0	600	400	

Figure 65: A Two-Level Executable Matrix of Alternatives

to vehicle-level elements.

Because it is assumed that the selection of vehicle-level scenario elements cannot affect program-level scenario elements through the relationship matrix, the forecasting problem can be treated as separate executable matrices of alternatives; an example of this is visualized in Figure 65. In this step, model parameters should be hierarchically organized into sub-matrices which have related parameters. Each sub-matrix can be treated as an independent problem with its own internal set of relationships; furthermore, one-way relationships originating from each sub-matrix can only affect other sub-matrices at a lower level in the hierarchy. In Figure 65, the upper executable matrix of alternatives is at a higher level and can have one-way relationships affecting the lower matrix.

This sub-step ends when the model parameters have been organized into sub-matrices. For some problems, this step will result in a single matrix if all model parameters have a first order effect and can be related to every other parameter via two-way relationships. For most problems, this step will end with two or more sub-matrices each with its own

internal relationship matrix. The next two sub-steps involve identifying one-way and two-way relationships.

6.2.3.2 Identifying One-Way Relationships

With the problem divided into two or more executable matrices of alternatives, the next step is to identify the one-way relationships between conditions of *different matrices*. Because these matrices should be organized in a hierarchal form where higher-level matrices influence lower-level matrices, for the purposes of this subsection, the higher-level matrix will be referred to as the upper matrix while the lower-level matrix will be referred to as the lower matrix.

This identification of one-way relationships should be done in two steps. First, the parameters of the upper matrix should be compared with every parameter in the lower matrix to determine if a relationship might exist. This reduces the analytical effort required since the number of comparisons between parameters is far less than the number of comparisons between individual conditions. For each parameter pair, it must be determined if a potential relationship exists between the parameters. If and only if a potential relationship exists, the parameter pair is brought forward into the next phase of this step.

Once the unnecessary parameter pairs have been eliminated, the analysis can focus on parameters pairs for which relationships will exist. One-way relationship matrices, as defined in Section 5.2.1.3 can be created for each parameter pair. After the one-way relationship matrices are created, the focus moves to identifying the relationships between conditions. As with the previous steps, the program office should leverage previous trade studies and risk mitigation studies to inform the creation of one-way relationships. Expert elicitation can also be used to augment existing knowledge or fill in any potential gaps. Some matrix elements will contain default ‘relationship does not impact’ values for potential relationships. For example, a baseline choice in an upper matrix will not have impact the baseline value in a lower matrix. The result of this sub-step will be the identification of all relevant one-way relationships between relevant parameter pairs.

6.2.3.3 Identifying Two-Way Relationships

The final substep is the identification of two-way relationships within each sub-matrix. This will involve completing a relationship matrix (Section 5.2.1.2) for each sub-matrix. A cross-consistency assessment must be performed to identify continuous incompatibilities and assess all effectual relationships. This analysis can be informed through expert elicitation and utilizing results of previous studies.

As with traditional MA, a cross-consistency assessment is performed to address incompatibilities. The relationship matrix implements this by giving incompatible pairs a probability relationship of 0 and fully compatible pairs a probability relationship of 1. Additionally, a continuous compatibility is implemented through the relationship matrix: values between 0 and 1 mean that two conditions are less likely to be chosen together while values over 1 mean that two conditions are more likely to be chosen together.

The final analysis of this step is to identify effectual relationships in each relationship matrix. Unlike one-way relationships where one is analyzing the effects of a higher-level condition on a lower-level condition, two-way effectual relationships involve mutual effects. Two-way relationships describe combinations of conditions which, when selected together, are mutually more mass efficient or mass inefficient.

6.2.4 Analysis

The first four steps of EMA concentrated on building the EMA model. The final step of EMA is to conduct analysis on this model to extract information. Three types of analysis can be conducted on this model. First, a probabilistic analysis can be conducted in order to extract relevant weight growth trends. The outputs of this probabilistic analysis can be used to inform a sensitivity analysis. Finally, this data can be analyzed to play what-if games by creating alternative development scenarios.

6.2.4.1 Probabilistic Analysis

The goal of probabilistic analysis is to extract a CDF of likely weight values. This will enable decision makers to decide how much margin to carry forward in the design by looking at

the CDF and specifying a percentile value.

As discussed in Section 5.4.3, the summary evaluation of an executable matrix of alternatives should only take place under certain circumstances— only small matrices with limited relationships should be fully evaluated. The summary evaluation will produce a probability mass function which can be integrated to form a CDF.

For most problems, a Monte Carlo simulation as described in Section 5.4.4.1 can be used. However, if the problem is organized into a hierarchy of matrices as described in Section 6.2.3.1, then a different procedure should be followed. This procedure involves executing the Monte Carlo process sequentially. A single Monte Carlo run will involve randomly selecting conditions from the highest level matrix followed by the next matrix in the hierarchy; this process continues until conditions have been selected from all matrices in the hierarchy.

6.2.4.2 Sensitivity Analysis

Another analysis which can be performed on the Monte Carlo simulation output data looks at the sensitivity of the weight predictions to each parameter. This allows analysts and decision makers to better understand which parameters and condition selections most influence the predicted weight. There are many ways to do this such as calculating the main effects or conducting a multivariate regression.

The main effect of a factor is defined as the difference between the averages of the response at the extreme settings of the response; larger main effects will have larger effects on the overall response. [90] In order to perform a main effects analysis on Monte Carlo simulation outputs, first the average response (total weight) for each condition in the matrix needs to be calculated. Knowing these average weights, the conditions in each parameter which provide the highest and lowest average response should be readily apparent— the main effect of each parameter is simply the difference between the highest and lowest average response.

Another analysis method which enables sensitivity analysis is multivariate regression; this analysis should be performed using statistical analysis software such as JMP. Multivariate regression will create an equation in which the Monte Carlo output weights are

functions of the EMA model input parameters and conditions. In a regression, the input parameters and conditions should be treated as categorical variables. While direct calculation using the EMA model is best for calculating individual combinations, the regression allows visualization of main effects and, more importantly, the visualization of slopes and derivatives. This information is displayed through a prediction profiler which visualizes the total derivative of each regression. [117]

6.2.4.3 What-If Scenarios

The sensitivity analysis allows decision makers to better understand which parameters and which conditions influence the overall output of the model. However, in order to make use of this information, decision makers need to play what-if games with the data. These what-if games enable decision makers to select specific alternative scenarios of development and immediately see results. This analysis is most useful when looking at which scenarios present the greatest impacts on weight growth. While adding design margin is one method of mitigating uncertainties, program managers have other methods of mitigating uncertainties such as additional funding or development time; EMA allows decision makers to see the prospective benefits of alternative mitigation factors before committing to a specific course of action.

The specific methodology for performing a what-if analysis is a filtered Monte Carlo. [15] In a filtered Monte Carlo, data filters are applied to the output of a Monte Carlo simulation. For the case of an EMA model simulation output, filters could include specific cutoffs of weight or selecting certain conditions. For example, an analyst may want to look at the expected value after excluding the top 5% of weight responses in order to re-evaluate the expected value of the sample. Another example would be excluding all cases which involve the selection of a certain condition which greatly influences weight. After the necessary filters are applied, probabilistic analysis should be re-run in order to recalculate the statistics of interest.

If a risk/opportunity workshop plans to use these techniques, then additional Monte Carlo simulation cases will be necessary. While Section 5.4.4.4 shows that a standard

probabilistic analysis can be adequately completed with approximately 1,500 to 2,000 cases, a multiple of this number should be run so that most case-exclusion scenarios will still yield ample cases from which to complete a probabilistic analysis. Approximately 10,000 cases should be a good number for most problems.

6.3 Combined Margin Analysis

The output from both range estimating and EMA will be a forecast of the dry weight of a vehicle. This can be converted into a margin by subtracting the vehicle baseline weight as defined by the WBS as seen in Equation 27; this equation holds for both the output of range estimating and EMA because both models should be calibrated versus the original WBS. Similarly, if a subsystem margins are desired, the subsystem outputs of both range estimating and EMA can be converted to subsystem level margins by subtracting the subsystem baseline from the model output as seen in Equation 49.

$$Margin_{Vehicle} = Weight_{ModelOutput} - Weight_{Baseline} \quad (27)$$

$$Margin_{Subsystem} = Weight_{SubsystemModelOutput} - Weight_{SubsystemBaseline} \quad (49)$$

Once margins have been extracted from the model output, the final step of the hybrid methodology is to combine the outputs from both range estimating and EMA to decide on an overall program margin. This can be done by allocating a separate margin during EMA and range estimating and combining the two at a top level. Another option would be to combine the underlying simulation data before performing data analysis.

If decision makers decide to allocate a margin based on the independent results of EMA and range estimating, then the final step of this method is very straightforward. In this case, decision makers would set separate margins for in-scope growth based on the results of range estimating and out-of-scope growth based on the results of EMA. This would be analogous to setting separate margins for mass margin and mass growth allowance in the AIAA Mass Properties Control for Space Systems Standard. [6] Another potential option for decision makers would be to allocate individual subsystem margins independently.

Decision makers also have the option of combining the resulting simulation data. In this case, the Monte Carlo simulation data from both range estimating and EMA would have to be converted into the predicted margins. This converted data can then be added together to get a single distribution for total predicted margin; because this involves combining two separate Monte Carlo simulations, it is recommended to perform additional simulation cases in order to guarantee a good sample size. Finally, an overall margin can be determined by looking at the output data. If decision makers wish to allocate based on individual subsystems, then the results of both simulations will have to be added together based on the identified subsystems from Section 6.2.1.1.

CHAPTER VII

SPACE SHUTTLE ORBITER STUDY

In order to fully substantiate hypotheses 1 and 2, the hybrid methodology comprised of EMA and range estimating must be demonstrated as able to predict the final weight of a novel vehicle. However, this form of substantiation is very expensive and difficult as it would require a large development program. In the absence of a novel development program willing to implement this method, the substantiation of hypotheses 1 and 2 will take the form of two experiments carried out in this chapter and Chapter 8.

This chapter focuses on the ability of EMA to evaluate prospective baseline changes; specifically, the baseline changes and their corresponding, prospective weight impacts. Because the focus is on EMA, range estimating will not be performed for the shuttle; instead, a predetermined percentage will be used to represent in-scope weight growth. If EMA can successfully “predict” the Space Shuttle Orbiter final weights based on early 1970s data, then this will provide evidence that EMA is a forecasting tool and help substantiate hypotheses 1 and 2.

7.1 Space Shuttle Orbiter Design Changes

7.1.1 ATP Orbiter

The NASA design of MSC-040C was the official design of record when the Orbiter program was granted the authority to proceed in March of 1972. [61] While this design would undergo many changes before production, it would serve as a basis from which future design changes were made. The ATP orbiter was projected to have a dry weight of 170,000 lbs including margin or 156,000 lbs not including margin. [61, 96, 95, 97]

The ATP orbiter, Figure 66, was built around a 3,220 sq. ft. blended delta wing with a leading edge sweep of 50°. This delta wing would allow a 170,000 lb Orbiter carrying a 40,000 lb payload to have a landing speed of 150 kts. Furthermore, this wing had to have a suitable hypersonic L/D ratio to accommodate Air Force requirements to land at

Vandenberg Air Force Base. Additional design considerations included hypersonic trim and reentry heating. The straight trailing edge contained elevons, and a body flap provided pitch control while shielding the main propulsion system from reentry heating. [61, 56]

Two additional propulsion systems were also included in the ATP Orbiter: air-breathing engines and abort solid rocket motors (ASRM). Two air-breathing engines were placed in the Orbiter's payload bay; the engine inlet was located at the base of the vertical stabilizer, and the nozzles were located just below the main propulsion system. This system was sized to be used during nominal reentry and ferry missions. The ASRM system consisted of two 30 ft long, 386,000 lb thrust solid rocket motors mounted on the side fuselage of the orbiter just above the wing. In the event of a failure in the early phases of flight, the motors would be used to boost the Orbiter to an altitude from which it could safely glide back to a runway near the launch site. Nominally, the ASRMs would be jettisoned from the Orbiter; later studies examined a nominal burn of these motors for additional impulse.[61, 56]

7.1.2 Program Readiness Review Orbiter

The Orbiter immediately underwent design refinement. The baseline design as of the program readiness review (PRR) can be seen in Figure 67. The orbital maneuvering system was moved from the side of the fuselage to the shoulders. The forward section of the fuselage was repackaged to improve aerodynamics. Additionally, small changes were made to the wing and wing/body fillet. [61]

The most significant changes at this stage were to the propulsion systems. The ASRMs were evaluated to determine if they could be used during boost, and concerns arose because deleting the ASRMs would require a redesign of the solid rocket boosters (SRB) to equip them with thrust vector controls. The trade study resulted in the decision to delete the ASRMs from the Orbiter because it was only useful for 30 seconds after ignition at the expense of added complexity; this added complexity added additional failure modes to the overall Space Shuttle stack which could result in a loss of mission. [56] In addition to the loss of the ASRM system, the air-breathing propulsion system was deleted from the Orbiter baseline. [61]

ATP CONFIGURATION (VEHICLE 1) MARCH 1972

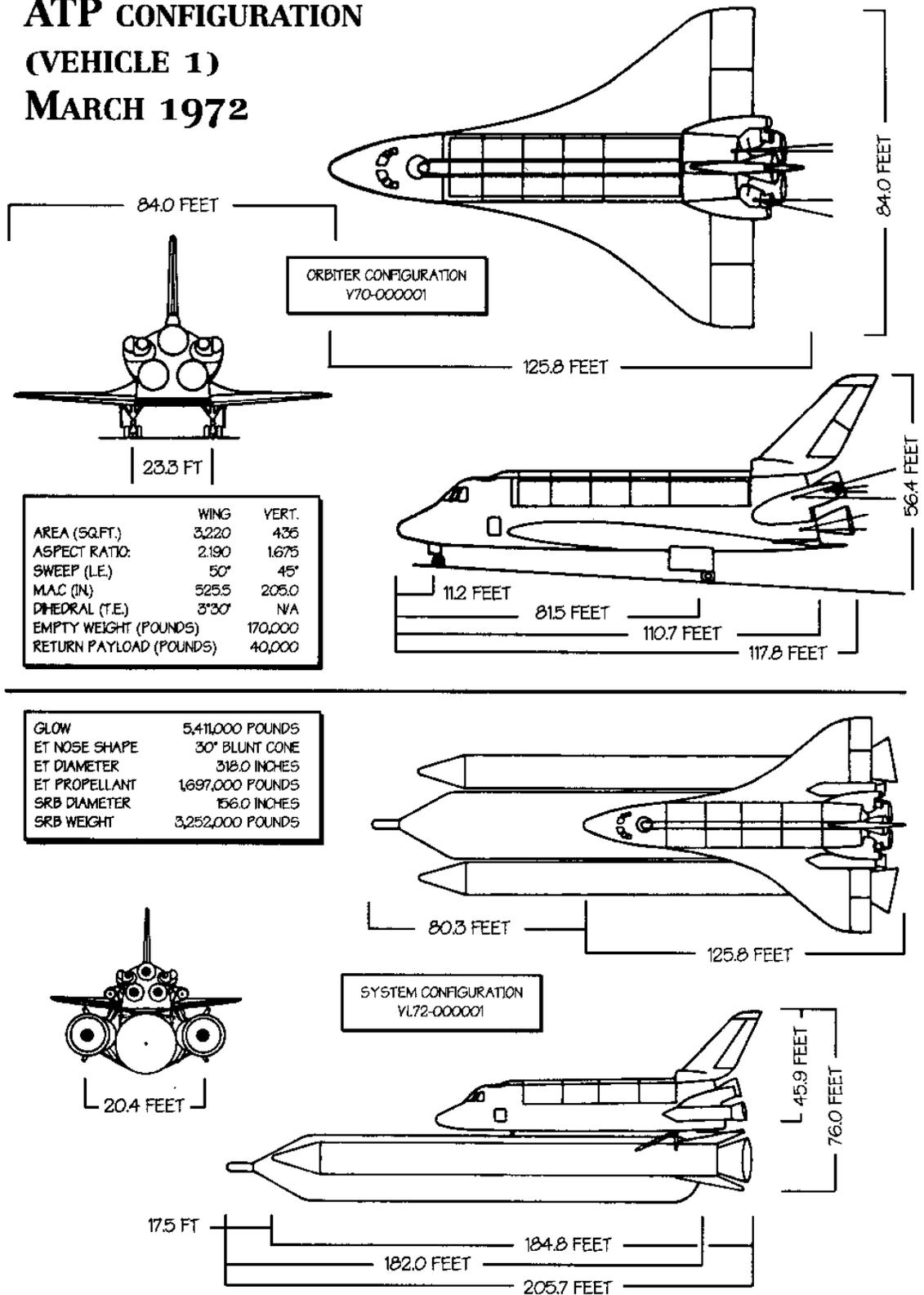
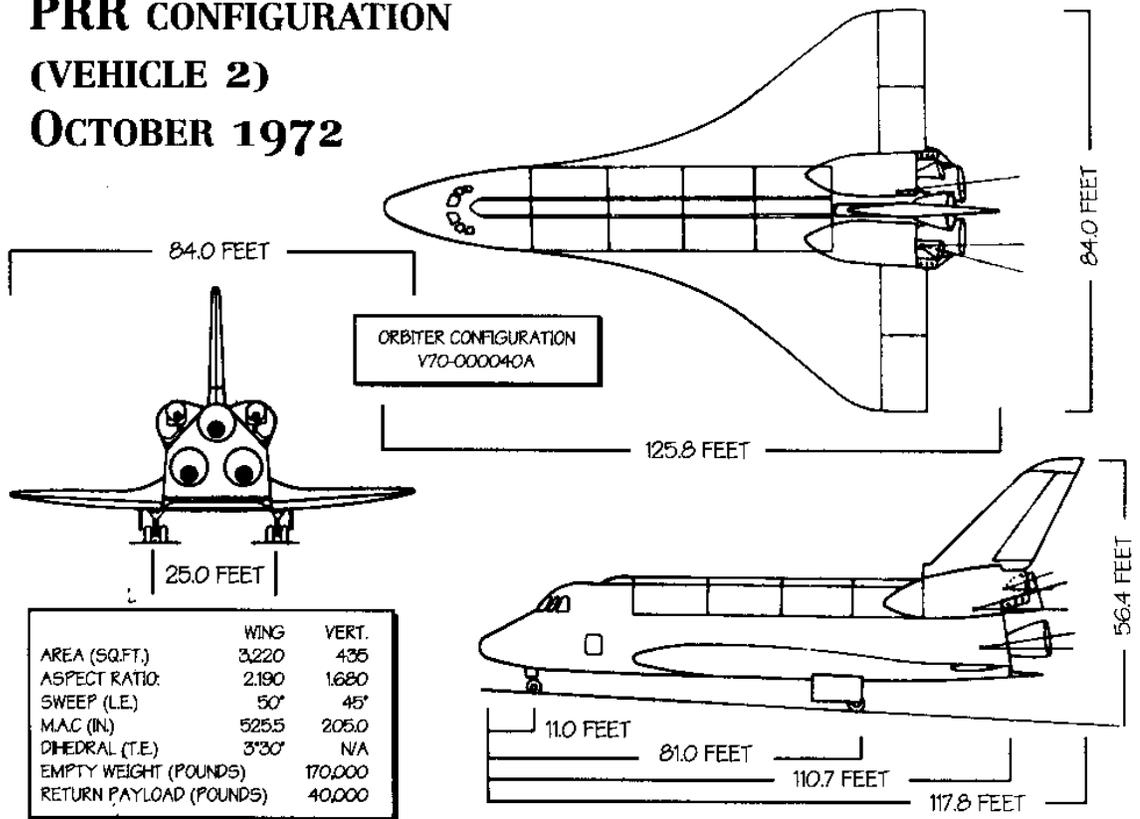
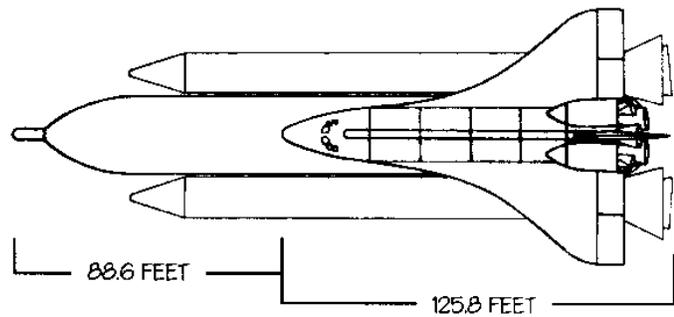
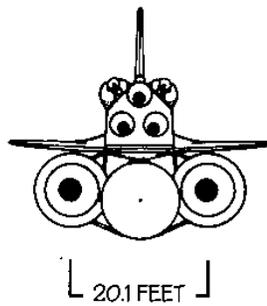


Figure 66: The Space Shuttle Orbiter at Authorization to Proceed [61]

PRR CONFIGURATION (VEHICLE 2) OCTOBER 1972



GLOW	5,261,000 POUNDS
ET NOSE SHAPE	568-INCH OGIVE
ET DIAMETER	304.0 INCHES
ET PROPELLANT	1,650,000 POUNDS
SRB DIAMETER	162.0 INCHES
SRB WEIGHT	3,276,000 POUNDS



SYSTEM CONFIGURATION
VL72-000030

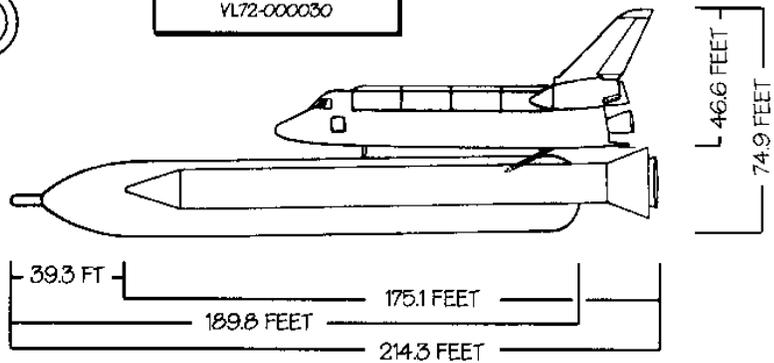


Figure 67: The Space Shuttle Orbiter at its Program Readiness Review [61]

7.1.3 150K Orbiter

The most substantial evolution of the Orbiter occurred after PRR as weight and cost savings played a more important role in the design decisions governing the Orbiter baseline. The previous two designs were expected to weigh 170,000 lbs including growth, but a series of redesigns were able to save an expected 20,000 lbs resulting in what is historically known as the 150k Orbiter. This new baseline can be seen in Figure 68. [61]

This substantial weight savings was driven by a complete redesign of the wing planform. While the main focus of Orbiter development was on the blended delta wing, NASA had also been studying the prospects of using a double-delta wing on the Orbiter. After PRR, it was decided that the double-delta wing would offer superior performance. The new double-delta planform had 79° and 45° sweep on its inner and outer sections respectively. This new wing had an area of 2,690 sq. ft.— 5/6ths the size of the original blended delta. This smaller wing drove a large reduction weight but required a relaxation of the landing speed requirement; the Orbiter now touched down at 165 knots up from 150 knots. [61, 56]

In addition to significant aerodynamic design changes, both the air-breathing engines and ASRMs returned to the design. The ASRM system was brought back by adding provisions to carry two ASRMs on the side of the Orbiter. The air-breathing propulsion system was very different from earlier Orbiters. On the 150k orbiter, five TF33-P7A turbofans and a payload-bay mounted fuel tank would only be installed for ferry flights. While the actual turbofans would not be accounted in the Orbiter dry mass, the hard points and plumbing would need to be carried into space. [61]

7.1.4 Vehicle 3/4 Orbiter

After the 150k orbiter redesign, the Orbiter's design continued to evolve throughout 1973. The biggest change involved the deletion of a docking hatch from the baseline; this deletion also resulted in a shortening of the fuselage by 3 ft. The aerodynamics design of the Orbiter also under went iteration. The sweep of the forward section of the double-delta wing was increased to 81° , and other changes were made to the incidence angle, airfoil shape, and wing mounting position to optimize for aerodynamic and aerothermal performance. Finally

VEHICLE 2A CONFIGURATION (150K ORBITER) DECEMBER 1972

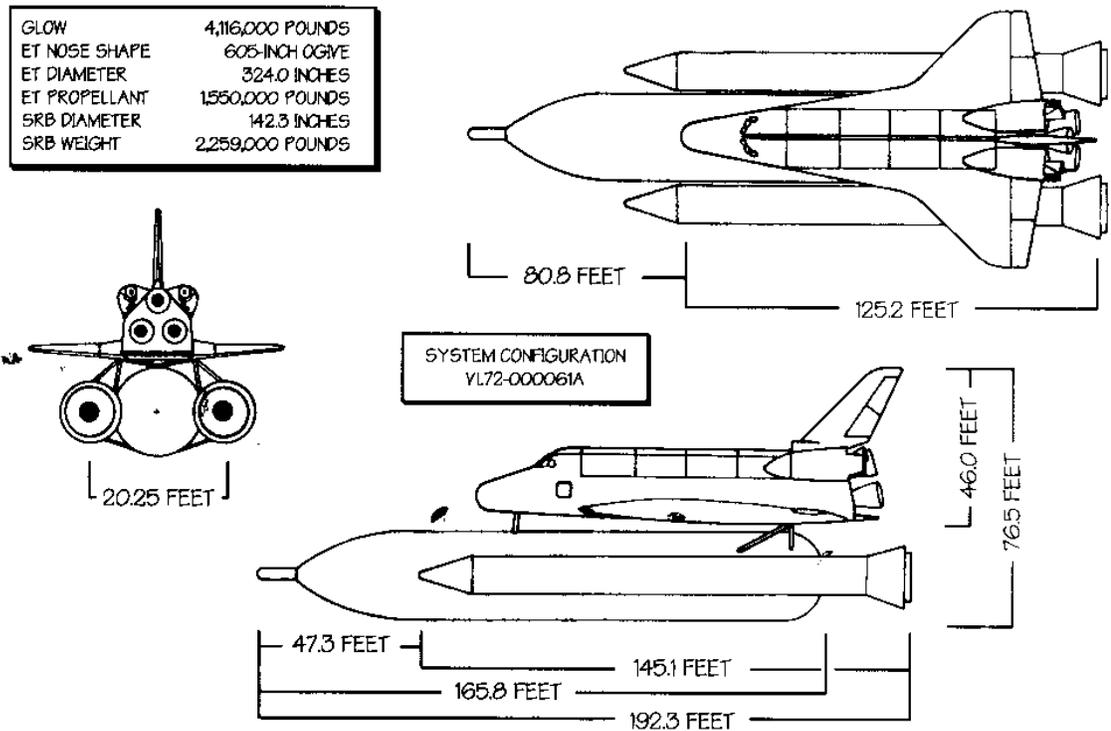
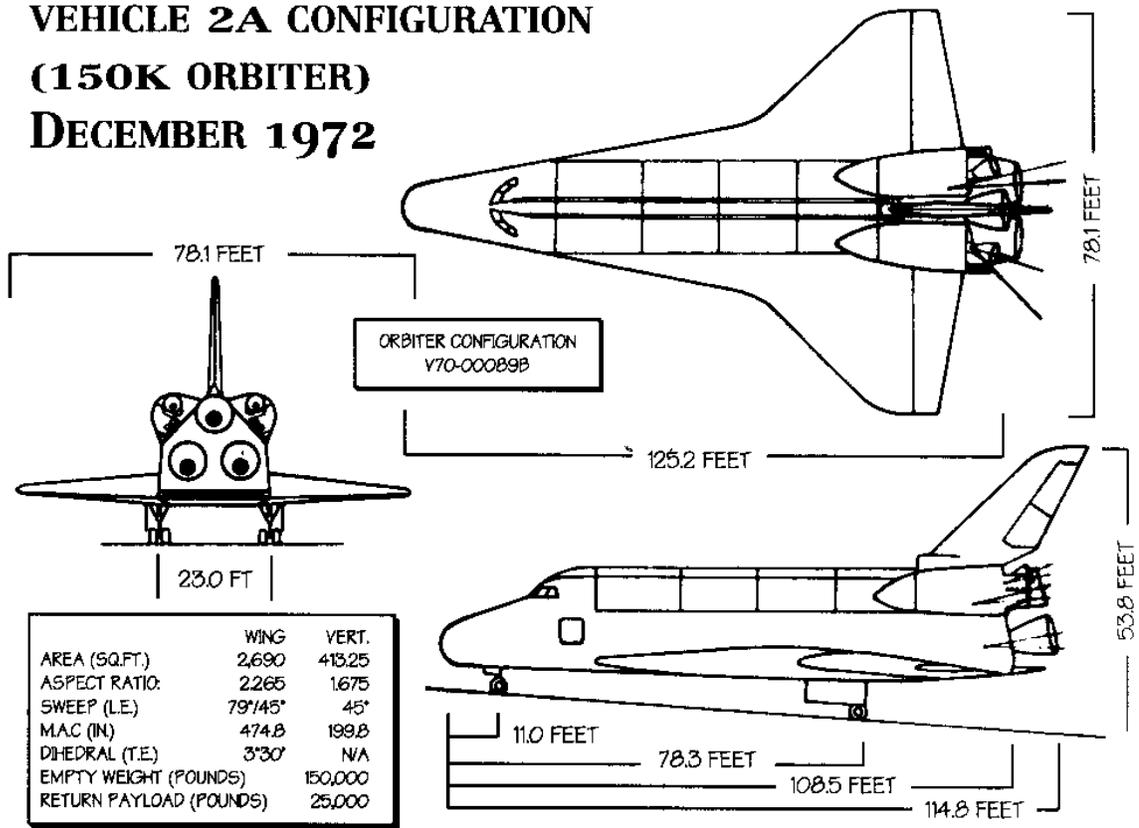


Figure 68: The 150k Orbiter [61]

the robotic arm was moved from its own fairing to be packaged in the payload bay. This updated Orbiter can be seen in Figure 69. [61, 56]

7.1.5 Vehicle 5/6 Orbiter

The design of the Orbiter continued to evolve throughout 1974. Most work consisted of mostly minor design changes compared to the previous redesigns. The biggest change to this Orbiter was a redesign of the OMS pods which enabled the payload bay door to be simplified from a 4 piece door to a single piece door. The forward fuselage was tweaked resulting in a slight reduction in fuselage length, a redesign of the forward RCS system, and the removal of doors meant to shield the forward RCS system during launch and reentry. The ASRMs were deleted again saving \$300 million in development cost, and the provisions for the air-breathing propulsion system were removed from the design. In order to reduce complexity and weight, a redundant hydraulic system was removed bringing the total number of hydraulic systems from four to three. Finally, a parachute was added to reduce the landing field length. [61]

7.2 Analyzing the Space Shuttle Orbiter

The Orbiter makes an ideal sample problem to demonstrate EMA– it was a novel concept which experienced a large number of design changes throughout its early development. Furthermore, original mass reporting documents exist to provide impact estimates of historical design changes. These historical design changes also represent potential weight savings and can demonstrate the method’s applicability to not only downside risks but also upside opportunities.

This study will start at the Orbiter’s ATP and try to project the final flight weight. The Orbiter’s ATP occurred in early 1972 before the contract award to Rockwell. [56] Because this occurs before contract award, the study will simulate an internal NASA study which could be used to inform requirements generation and derive the ultimate program weight requirement.

The first section of this study involves estimating baseline-specific growth. A pre-determined percentage will be used so that the overall probabilistic results at the end of

VEHICLE 3/4 CONFIGURATION MAY 1973

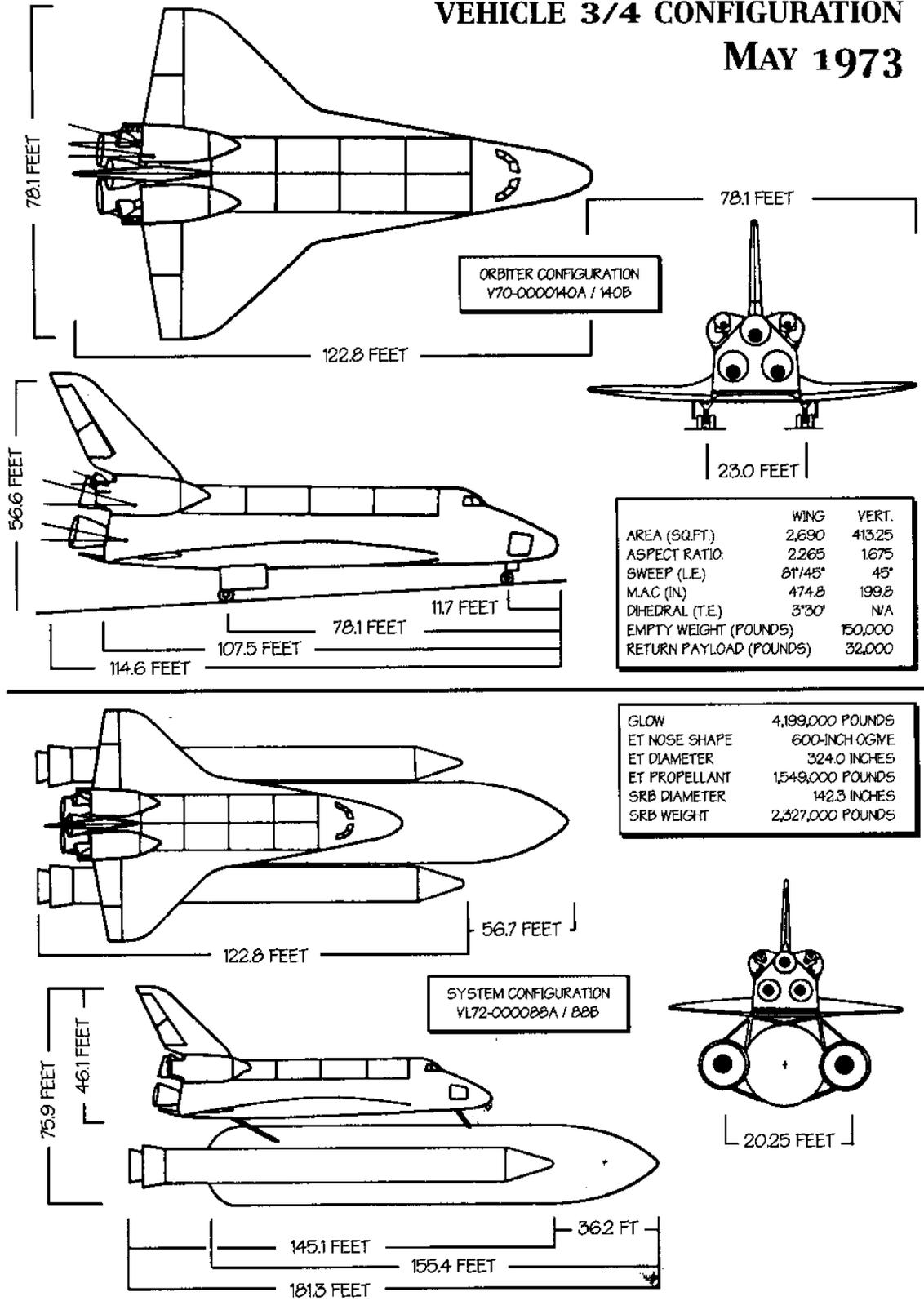
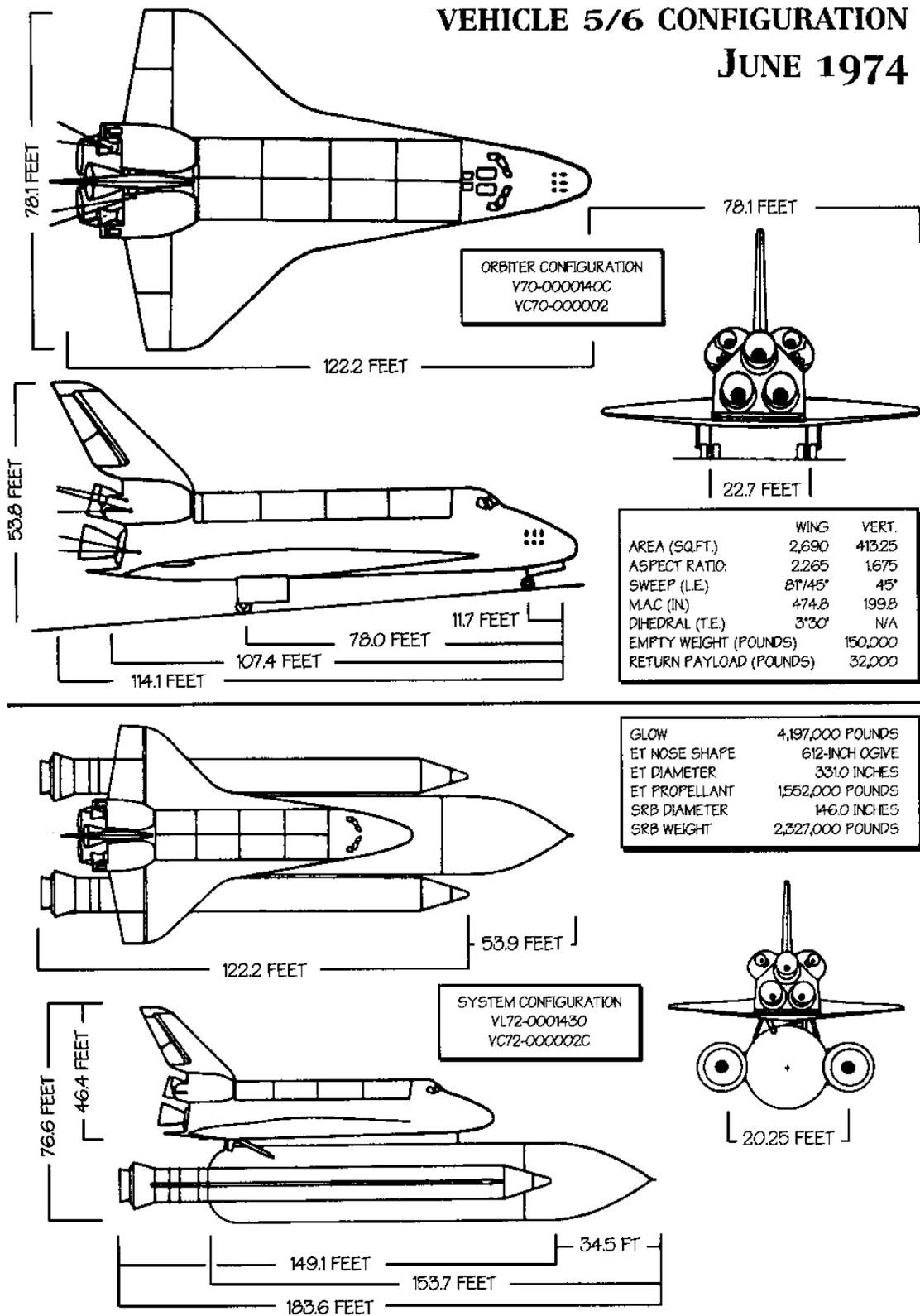


Figure 69: The Vehicle 3/4 Orbiter Configuration [61]

VEHICLE 5/6 CONFIGURATION JUNE 1974



the study will only reflect the output of EMA. The second section walks through each step of EMA as applied to the Orbiter problem. Finally, the last section discusses the results of the study.

7.2.1 Baseline-Specific Growth

While this sample problem primarily focuses on forecasting the impact of the Orbiter's baseline changes, the overall hybrid methodology requires an estimate of baseline-specific growth. In order to isolate the effects of EMA's forecast on the weight forecast, a predetermined percentage is used to account for baseline-specific growth.

Because many margin standards call for a margin to be in place at ATP, a relevant predetermined percentage compatible with this level of design fidelity must be used. The AIAA Mass Properties Control for Space Systems provides recommended allowances for in-scope growth as a function of design maturity level. The corresponding maturity level for ATP is roughly equivalent to an early conceptual design. The recommended value for large subsystems such as structures and propulsion is 15% while other subsystems such as electrical components, wire harnesses, and instrumentation can range up to 30%. [6]

The original Orbiter's dry weight at ATP was estimated to be 156,000 lbs. [96, 95, 97] To account for baseline-specific growth, a 20% margin is applied. A 20% margin is selected based on the ranges of the AIAA recommended values— the 20% margin is large enough to account for both large subsystems and electrical components while not over-allocating due to the margin recommendations for lighter subsystems. The final result is an Orbiter ATP weight allocation of 187,200 lbs— a 31,200 lb margin.

7.2.2 EMA

7.2.2.1 Identification of Model Parameters

For the Orbiter example problem, the model parameters used in EMA should reflect the historical record of baseline changes. While this problem could include other parameters which are not reflective of historical design changes, only historical design changes are considered because mass estimates for each design iteration can be derived from historical Orbiter mass properties reports.

The morphological field used for this problem can be seen in Figure 71. In order to capture the largest design changes to the Orbiter, the morphological field contains parameters for modeling the air-breathing engine subsystem, the ASRMs, the wing planform, and the fuselage length. Additionally, the morphological field includes some of the smaller additions and subtractions by including the number of hydraulic systems and a drag chute for landing.

7.2.2.2 Identification of Conditions

The air-breathing engines went through different designs during the evolution of the Orbiter. The three air-breathing conditions in Figure 71 represent the three distinct design options developed for this system: system non-inclusion, a payload bay mounted propulsion system, and wing mounted engines for ferry missions. The figure indicates that the original ATP Orbiter had payload bay mounted air-breathing engines.

The ASRM system was another highly-debated subsystem in the Orbiter. The morphological field in Figure 71 shows that ASRMs can either be on the Orbiter or deleted from the system. The highlighted “two” condition reflects the ATP baseline.

The Orbiter had two separate wings during the course of its development: the 50° blended delta and the double-delta. Both of these options are listed as possible choices in Figure 71 with the baseline blended delta wing highlighted as part of the baseline.

Figures 66 through 70 list four separate fuselage lengths. The drop from 125.2 ft to 122.8 ft was the most significant drop which occurred when the docking hatch was removed from the Orbiter, but two other reductions were made in the process of design maturation. All four options are included in Figure 71; the option for 125.8 ft is highlighted to indicate the original ATP Orbiter baseline.

Two of the changes made in the later stages of design are also included in Figure 71. The option of reducing the number of redundant hydraulic systems and adding a parachute are included in this model. The four redundant hydraulic systems and a lack of drag chute is highlighted in Figure 71 as the ATP Orbiter baseline.

Air-Breathing Engines	None	Payload Bay Mounted	Wing Mounted
ASRM	None	Two	
Wing Planform	Blended Delta	Double-Delta	
Fuselage Length	122.2 ft	122.8 ft	125.2 ft 125.8 ft
Hydraulic systems		3	4
Parachute	none	Single	

Baseline

Figure 71: Space Shuttle Orbiter Design Change Morphological Field

7.2.2.3 Identification of Condition Attributes

A key reason for selecting the Orbiter as a sample problem was the existence of historical mass properties reports. For the task of identifying the effects for each condition, the original documents provide a key aspect of this study– the elimination of hindsight bias. This will allow the study to effectively “predict” the final flight weight based on original estimates. Therefore, when possible, the condition effects will be derived from these original documents. A graph of the original CBE Orbiter mass estimates from July 1973 can be seen in Figure 72.

For this study, each condition has a probability and an additive effect. Multiplicative effects are omitted because each parameter identified in Section 7.2.2.1 corresponds to a specific change to a portion of the vehicle and is not a vehicle-wide effect. While probability values would normally be derived from expert input at a risk/opportunity workshop, the author will act as a surrogate for a workshop and assign probabilities for each condition. As a general rule, the baseline conditions always have at least a 25% chance of occurring, and options which save mass are assigned higher probabilities. These higher probabilities

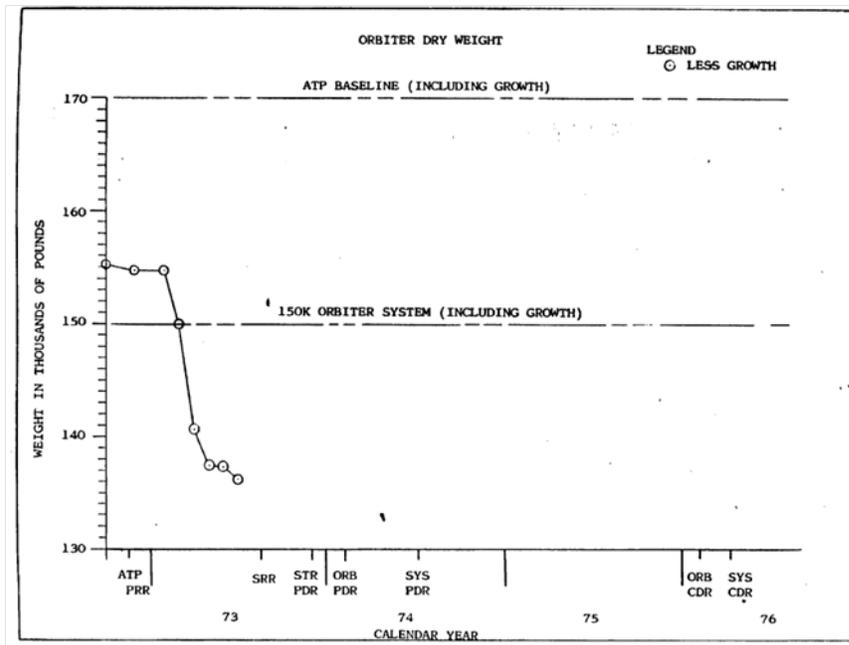


Figure 72: Space Shuttle Mass Changes as of July 1973[95]

are meant to simulate the need to save mass from early Orbiter designs.

Figure 72 consists of three large drops in weight during the calendar year 1973. The first drop from approximately 155,500 lbs to 150,000 lbs corresponds to incorporating the deletion of the air-breathing engines and ASRMs into the official mass estimates. The second drop in weight corresponds to the 150k orbiter redesign, this brought the CBE to approximately 140,000 lbs. The next 3,000 lb drop in mass should correspond to the deletion of the Orbiter’s docking hatch and the reduction of fuselage length by 3 ft. [95]

The deletion of air-breathing engines is assumed to account for most of the drop from a 155,500 lb Orbiter to a 150,000 lb orbiter in early 1973 in Figure 72. Based on this information, the difference between the payload by mounted propulsion system and the complete deletion of the air-breathing engines is assumed to account for a weight savings of 4,000 lbs. Because the wing-mounted system only requires additional hard points and plumbing, it is assumed to only cost 250 lbs compared to the complete removal of the system. The ATP baseline is assumed to have a 25% chance of occurring. The complete removal of the system represents a large weight savings and is assumed to have a 50% chance of

selection. Finally, the wing-mounted option claims the remaining 25% probability. These values are summarized in Figure 73.

Because the air-breathing propulsion system takes up 5,000 lbs of the 5,500 lb drop, the ASRM system is assumed to occupy the remaining 500 lbs. This 500 lbs would account for the additional structure, hard points, and mechanisms related to the ASRMs— as the ASRMs are jettisoned, they would not be accounted for in the Orbiter's dry weight statement. The baseline option is assumed to have a 25% probability of selection while the weight-saving deletion of the ASRM system is assumed to have a 75% probability of selection.

The wing planform has two conditions: the original blended delta and the 150k double-delta. Because the blended delta represents the original baseline, the effect value is 0 lbs, and it is assumed to have a 25% probability of occurring. The double-delta is assumed to represent the 10,000 lb reduction in weight as seen in Figure 72, therefore has an effect of 10,000 lbs of weight reduction. A 75% probability is assigned to this because it represents a likely design change due to the NASA and Lockheed double-delta studies. [61]

The immediate drop of 4,000 lbs in mid-1973 is assumed to account for the shortening of the fuselage by 2.4 ft and the elimination of the docking hatch. Of these 4,000 lbs, 3,000 lbs are assumed to be fuselage related, and 1000 lbs are assumed to be part of the docking mechanism. Based on these assumptions, the forward fuselage can be calculated to approximately weigh 1,250 lbs per foot. This assumption allows the effects in fuselage length to be approximated. First the 125.8 ft fuselage has a 0 lb effect as it represents the baseline. The first reduction from 125.8 ft to 125.2 ft is a reduction in 0.6 ft and assumed to be a reduction of 750 lbs relative to the baseline. From there, the reduction of 4,000 lbs as noted in Figure 72 occurs leading to an effect of a total reduction of 4,750 lbs. The final option in fuselage length is a reduction to 122.2 ft; this 0.6 ft reduction will be assumed to weigh 750 lbs less than the 122.8 ft fuselage length option resulting in a total reduction of 5,500 lbs relative to the baseline.

According to the ground rules for this experiment, the baseline condition for fuselage length is assigned a probability of 25%. Because the reduction to 122.2 ft and 122.8 ft represents significant weight savings, the probabilities are biased in their favor as can be

Air-Breathing Engines	None P(0.5), Effect = -5000 lbs	Payload Bay Mounted P(0.25), Effect = 0 lbs	Wing Mounted P(0.25), Effect = -4750 lbs	
ASRM	None P(0.75), Effect = -500 lbs	Two P(0.25), Effect = 0 lbs		
Wing Planform	Blended Delta P(0.25) Effect = 0 lbs	Double-Delta P(0.25), Effect = -10000 lbs		
Fuselage Length	122.2 ft P(0.35) Effect = -5500 lbs	122.8 ft P(0.25), Effect = -4750 lbs	125.2 ft P(0.15) Effect = -750 lbs	125.8 ft P(0.25), Effect = 0 lbs
Hydraulic systems	3 P(0.5) Effect = -400 lbs	4 P(0.5), Effect = 0 lbs		
Parachute	None P(0.5) Effect = 0 lbs	Single P(0.5) Effect = 500 lbs		

Baseline

Figure 73: Space Shuttle Orbiter Design Change Executable Matrix of Alternatives

seen in Figure 73. Finally, the reduction to 125.2 ft is given a probability of 15%– the remainder of the available probability for the parameter.

The final two parameters in Figure 73 represent the changes made at the end of the Orbiter’s early design. Because of this, the probability is listed as 50% for each condition. As the number of hydraulics systems is reduced from three to four, the weight savings will be approximately 3/4ths of the July 1973 weight statement for hydraulics (400 lbs reduction). [95] Finally, the parachute system and associated structural strengthening to accommodate parachute loads are assumed to be 2/3rds of the SRB parachute weight (500 lbs addition).

7.2.2.4 Identification of Relationships

The next step in the creation of the EMA model is conducting a cross-consistency assessment and identifying any relationships between conditions. Because this is a simple model with each parameter modeling a disparate subsystem, each parameter will be assumed to be independent of one another– there are no incompatibilities in this model. Additionally, each condition is assumed to not influence the additive effects of every other conditions– there are no effectual relationships in this model. These assumptions simplify this analysis

Table 14: Orbiter Dry-Weight Predictions

	Expected Value Case	80% Case
Baseline Dry Weight	156,000 lbs	156,000 lbs
Baseline-Specific Growth	31,200 lbs	31,200 lbs
EMA Adjustment	-14,700 lbs	-10,250 lbs
Predicted Dry Weight	172,500 lbs	176,950 lbs

and produce a “blank” relationship matrix.

7.2.2.5 Analysis

Since the Orbiter study is a small problem with only 192 potential combinations, a summary evaluation of all simple combinations is possible. The CDF of this summary evaluation can be seen in Figure 74.

In this figure, two values are specifically called out: the expected value and the 80% value. The expected value is of interest because it is an important statistic of the distribution; the expected outcome of these potential baseline changes is a weight savings of 14,738 lbs (rounded to 14,700 lbs). Additionally, it can be seen that the expected value represents the 65.8% quantile. The 80% quantile is a recommended choice for choosing a conservative estimate in range estimating literature, therefore it is an interesting number to pull from the EMA CDF. [2] The 80% quantile prediction in this problem represents a weight savings of 10,250 lbs. Table 14 presents a summary of the Orbiter’s original dry weight, the baseline-specific mass growth allowance, the adjustment in mas due to EMA, and the final predicted dry weight for both the expected value and 80% quantile predicted adjustments.

7.3 Results and Conclusions

Because the potential baseline changes to the Orbiter represented a potential weight savings compared to the ATP Orbiter, the goal of EMA in this problem is to forecast the likely weight savings to the ATP Orbiter. To test EMA’s forecast, the final flight weight of each orbiter can be compared to the expected value and 80% probability forecast; the roll out weight, space shuttle main engine weight, and total dry weight for each orbiter can be seen in the first three rows of Table 15.

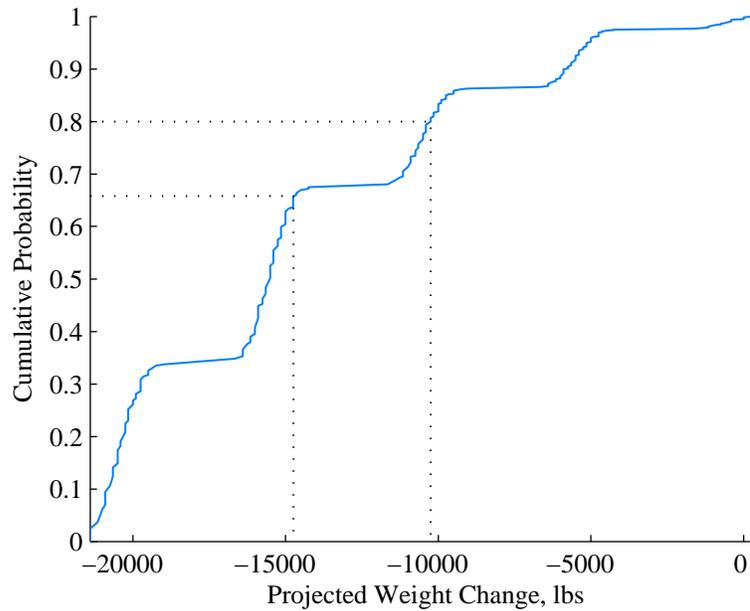


Figure 74: CDF of Projects Changes to Orbiter Weight

Combining the original ATP Orbiter dry weight estimate (156,000 lbs), 20% mass growth allowance (31,200 lbs), and the expected value weight savings (-14,700 lbs) yields a total predicted weight of 172,500 lbs. The fifth row of Table 15 shows both the absolute and percentage comparison of the expected value prediction to each orbiter flight weight. As can be seen in the table, the less conservative expected value predictions are off by a few percentage points on earlier orbiters (Challenger and Columbia), but very closely predict the final flight weights of the three later orbiters (Discovery, Atlantis, and Endeavour).

Similarly, combining the original ATP Orbiter dry weight estimate (156,000 lbs), 20% mass growth allowance (31,200 lbs), and the expected value weight savings (-10,250 lbs) yields a total predicted weight of 176,950 lbs. The last row of Table 15 shows both the absolute and percentage comparison of the expected value prediction to each orbiter flight weight. The more conservative 80% quantile prediction is more successful. This prediction closely predicts the first two Orbiters off of the production line within a single percent accuracy. While it does not closely predict the three later orbiters, the forecast does provide an ample design margin.

Table 15: Comparisons Between Flight-Weight Orbiters and Predicted Weights

Orbiter	Challenger OV-099	Columbia OV-102	Discovery OV-103	Atlantis OV-104	Endeavour OV-105
Roll Out Weight (lbs)[61]	155,400	158,289	151,419	151,315	151,205
SSME Weight (lbs)[91]	20,626	20,626	20,626	20,626	20,626
Total Dry Weight (lbs)	176,026	178,915	172,045	171,941	171,831
Predicted Weight of Expected Value (lbs)	172,500	172,500	172,500	172,500	172,500
Difference From Expected Value (lbs,%)	-3,526 -2.0%	-6,515 -3.6%	455 0.26%	559 0.33%	669 0.39%
Predicted Weight of 80% Quantile Prediction (lbs)	176,950	176,950	176,950	176,950	176,950
Difference From 80% Quantile Prediction (lbs,%)	924 0.52%	-1,965 -1.1%	4,905 2.9%	5,009 2.9%	5,119 3.0%

The results of this experiment show that EMA can be used to accurately predict changes in predicted weight due to baseline changes. This experiment used original design changes and the corresponding estimated weight savings from the Orbiter program to make a successful forecast of the final flight weight. This successful experiment substantiates EMA's use as a forecasting technique and substantiates hypotheses 1, 2, and 3.

CHAPTER VIII

FAST PROGRAM PROOF OF CONCEPT

The previous chapter illustrated EMA as applied to the historical example of the Space Shuttle Orbiter. While this experiment successfully predicted the flight mass of the Orbiter, it only partially substantiated hypotheses 1 and 2. In order to fully substantiate these hypotheses, a full demonstration of the hybrid methodology combining range estimating and EMA is needed. This chapter documents a sample problem from the Air Force Research Laboratory's future-responsive access to space technologies program.

8.1 Future-Responsive Access to Space Technologies Program

The future-responsive access to space technologies (FAST) program is a set of technology development studies initiated by the Air Force Research Laboratory (AFRL) in 2007. These studies covered three technology areas necessary for future boost vehicles: design and operability, advances structures and structural health monitoring, and adaptive guidance navigation and control. [12, 13, 135] These projects were awarded to Northrop Grumman, Lockheed Martin, and Honeywell, respectively. [12]

A central feature of the FAST program was that each technology development program would be coordinated with one another; each contractor would utilize a common reference flight system (RFS) as a basis for each individual program. This coordination is part of a phased development approach where the individual programs focus on both technology development and integration. [106]

The development of the FAST RFS was driven by vehicle-level measures of merit. These included rocket-back test flight performance, single stage vehicle performance, aerothermal capability, and operational goals such as call-up time and turnaround time. [135] After three design cycles, configuration F, shown in Figure 75, was agreed upon as a common baseline for the three technology development programs.

Because the FAST RFS F is a novel concept brought into a late conceptual design

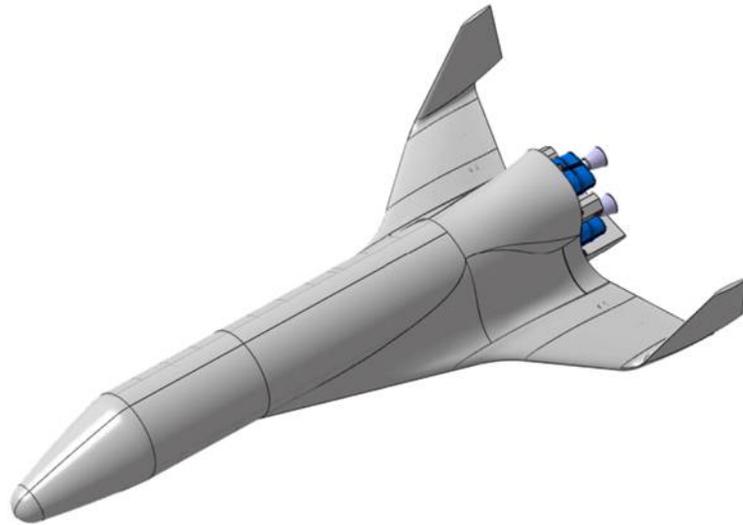


Figure 75: FAST RFS F [12]

level of maturity, it serves as a good sample problem for margin estimation using the hybrid methodology combining range estimating and EMA. This chapter details the analysis of the RFS F. Section 8.2 documents the range estimating analysis including the WBS generation, assignment of ranges, and simulation. Section 8.2 documents EMA as applied to the RFS F including the creation of executable matrices of alternatives, its corresponding relationship matrices, and the subsequent analysis.

8.2 Range Estimating

8.2.1 Work Breakdown Structure

The first step in conducting a range estimating analysis is determining a baseline WBS. For this problem, there are two sources of mass properties data: Northrop Grumman conceptual estimates and Lockheed Martin subsystem estimates. The Northrop Grumman estimates were derived during the RFS conceptual design process while the Lockheed Martin estimates were derived from the initial design of the advanced structures technology demonstrator. For this sample problem, the two sources of data were combined into a single WBS. Because Lockheed Martin estimates were indicative of the technology demonstration program, they were used in place of Northrop Grumman estimates wherever possible. The WBS used for

this sample problem can be seen in Table 16. [98]

8.2.2 Estimate Ranges for Individual Subsystems

The establishment of a baseline WBS provides a point estimate of subsystem mass properties. In order to conduct a range estimating analysis, weight engineers assigned to each subsystem would produce low, likely, and high weight estimates for each subsystem. However, as this is a sample problem to illustrate the utility of the method, the estimates by individual weight engineers will be replaced with percentage offsets based on the WBS subsystem point estimates. These percentage offsets will enable the generation of a defensible low, likely, and high estimate for FAST RFS F subsystems based on published margin recommendations.

The AIAA standard for space systems mass properties control recommends a mass growth allowance based on the type of subsystem and design maturity level. [6] The subsystems in the FAST RFS F WBS correspond to the following categories in the AIAA standard: structures, thermal control, propulsion, and electrical components. Furthermore, the FAST RFS F was developed to a design maturity level equivalent to layout drawings. These two pieces of information yield the AIAA standard's recommended margin. Structural and propulsion subsystems should receive 15% margin while thermal control and electrical components (assumed to be in the standard's 5-15 kg category) should receive 20% margin. [6]

Next, the AIAA recommended margins can be used to inform a low, likely, and high weight estimate which can then be used to create a triangular distribution. For each type of subsystem, the weight corresponding to the recommended margin is taken to be the high estimate. The most likely estimate will be assumed to be 5% less than the AIAA recommended margin. Finally, the low weight estimates for each subsystem are assumed to be 2.5% less than the original weight estimate. The percentage offsets for each subsystem type are summarized in Table 17. These offsets are applied to each subsystem listed in Table 16 to create the low, likely, and high numbers for the subsystem triangular distributions.

Table 16: FAST RFS F Weight Breakdown Structure

	Mass(lbm)	Mass(lbm)	Mass(lbm)
Structure	8891		
Wing Group		3831	
Exposed + Carry Thru			3240
Wing Fairing			382
Hot Elevon Structure			209
Vertical Fins		285	
Basic Vertical Fin			135
Rudder- Hot Structure			150
Body Group		3496	
Integral Fuel Tank			
Structure- Tank			662
Integral Oxidizer Tank			
Structure- Tank			867
Structure- Common Bulkhead			109
Insulation- Tank			252
Secondary Structures			
Forward Fuselage			442
Mid Body Fairings			355
Aft Fuselage			326
Thrust Structure			258
Body Flap			225
Landing Gear Group		1279	
Main Gear			1087
Nose Gear			192
Induced Environmental Protection	1161		
Wing Group TPS		276.2	
Tail Group TPS		67.1	
Body Group TPS		714.5	
Internal Insulation		79	
Hazardous Gas Detection		25	
Propulsion	2759		
Main Engines		1500	
Press + Feed System		890	
Gimbal Actuation		102	
Purge System		159	
Engine Heat Shield		108	
RCS	459		
Systems and Equipment	3810		
Prime Power System		888	
Electrical System		543	
Flight Control System		472	
Avionics System		611	
Flight Termination System		25	
Flight Test Instrumentation		600	
Environmental Control System		671	
Total Dry Weight	17080		

Table 17: Percentage Offsets for Range Estimating

Subsystem	Low	Likely	High
Structures	-2.5%	10%	15%
TPS	-2.5%	15%	20%
Propulsion	-2.5%	10%	15%
Electrical	-2.5%	15%	20%

8.2.3 Identify Dependencies

The next step of range estimating is to identify dependencies within the ranged subsystems. Ordinarily, this would involve performing a pairwise comparison to establish a correlation matrix. However, for this sample problem, a weight engineering team is unable to provide these correlations. In order to proceed with range estimating, the assumption of independence will have to be made. This assumption is reasonable because it does not substantially change the methodology used to analyze the RFS F; this assumption will only change the numeric output of the range estimating Monte Carlo simulation.

8.2.4 Simulation

A 10,000 case Monte Carlo simulation was performed over the range estimating model. The output of this model estimated the probable weight of the FAST RFS F baseline. A CDF of the probable weights can be seen in Figure 76. As with the shuttle study, the FAST RFS F study tracks the expected value and 80% quantile in the output of each simulation; these values are illustrated in Figure 76. The range estimating model predicts an expected value of 18,529 lbs and an 80% quantile level of 18,677 lbs.

While looking at the overall predicted mass is informative, the margin to apply to the vehicle is the quantity of interest. The original baseline weight of 17,080 lbs must be subtracted from the distribution in order to determine the margin due to design maturation; a CDF of this value can be seen in Figure 77. The model predicts an expected value of 1,448 lbs and an 80% quantile level of 1,596 lbs.

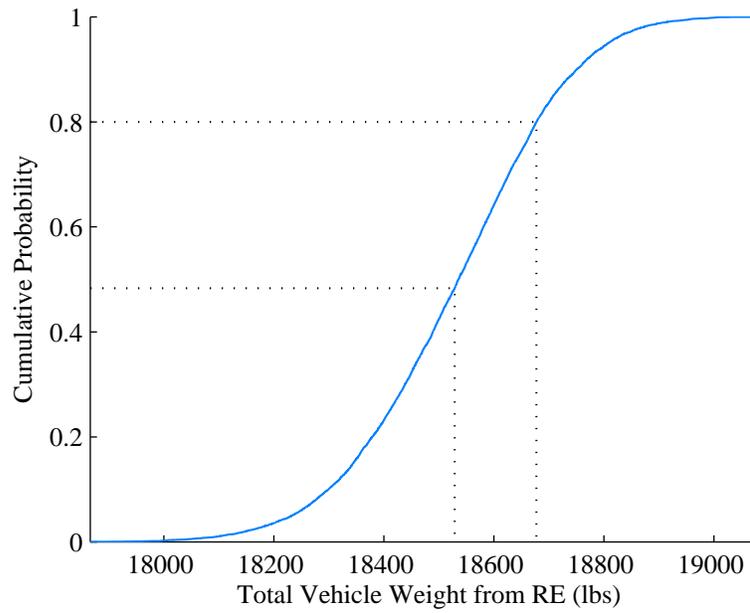


Figure 76: CDF of Total Vehicle Weight from Range Estimating

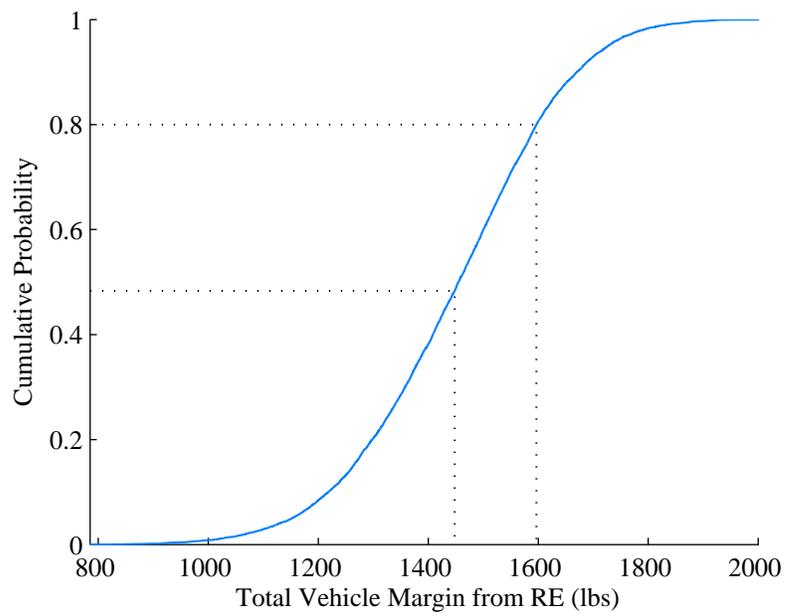


Figure 77: CDF of Total Margin from Range Estimating

8.3 *EMA*

8.3.1 Identification of Model Parameters

The first step in analyzing the FAST RFS F is to define the domains of the parameters needed to model the development program. All programs are subject to outside programmatic risks and opportunities, therefore the model should have parameters which reflect these programmatic issues. Additionally, the FAST program is technology-centric; the individual technology development programs which affect the dry weight should also be modeled. Finally, a physical decomposition can be used to model the vehicle. This subsection describes the individual parameters used in the completed morphological field which can be seen in Figure 78.

The first parameters which are identified describe the external influences on the development program which manifest exogenous uncertainties. The most obvious external influence which can influence a development program is the budget— this will affect the underlying technology development programs and the available engineering hours used by the program to optimize the design. In addition to the program’s budget, the requirements are also often degrees of freedom. For the FAST RFS F, three requirements are identified as parameters for the model. The first of these requirements is the aerothermal performance. One reference trajectory for the RFS F is a mission which stresses the TPS; adjusting this requirement will directly affect the sizing of the required TPS. [98] The next requirement describes the overall single-stage vehicle performance; this requirement sizes the overall vehicle. Finally, a transverse G-limit requirement determines the loads necessary for the sizing of structures and places a constraint on the reference trajectories.

The next set of parameters describe the technologies relevant to the dry weight of the vehicle and the corresponding technology uncertainty. Three structural technology development programs, part of the Lockheed Martin project, develop a reduced depth wing box, composite fuel and oxidizer tanks, and a composite common bulkhead. [12] Each of these three technologies is a parameter of the model. Finally, an additional parameter is used for mechanically fastened TPS materials. Technology programs are modeled independently to

model the progress of the development program. Each technology has corresponding parameters in the vehicle physical decomposition to model the integration of the technology on the final vehicle.

The vehicle parameters used in this model follow a physical decomposition of the vehicle which is driven by the WBS in Table 16; these parameters describe model uncertainties inherent in early-phase design which are not covered by the range estimating analysis.. The structural subsystems of the RFS F are modeled by a total of 17 parameters. The wing group is modeled by two parameters; one describes the wing shape/design while another describes load-case scenarios. The vertical tail is also described by two parameters; one parameter sets the location of the tail while another describes its construction. The elevon and body flap each receive their own parameters, and an additional parameter allows canards to be added to the vehicle. The body group is modeled by parameters which describe the fuel tank, oxidizer tank, and secondary structures. Both tanks have an identical parameterization; one parameter describes the load requirements, one parameter describes the structural design, and the third parameter describes the construction material. A separate parameter, tank attachment structure, models the common bulkhead or intertank. Finally, the nose structure, (mid-body and aft) secondary structures, and thrust structure each receive their own parameters.

An additional four parameters are used to model the TPS systems of the RFS F. Separate parameters are used for the wing group, tail group, and body group external TPS. Finally, an additional parameter is used to model the propellant tank thermal insulation. These four parameters were chosen in order to correspond to the WBS in Table 16; hazardous gas detection was omitted from this section because this subsystem would likely be made from off-the-shelf parts and subject far less uncertainty than the other four TPS parameters.

A final five parameters are used to model alternatives for the propulsion system. One parameter is used to model alternative engines which could be used to power the RFS F. Another parameter models the way that the engines are integrated into the vehicle's aft structure. A third parameter represents the different alternative methods of constructing the oxidizer feedline which connects the forward-mounted oxidizer tank to the main propulsion

Table 18: Organization of Parameters by Subsystem

Subsystem	Parameters
Propellant Tanks	Composite Tank Technology, Common Bulkhead Technology, Fuel Tank, Fuel Tank Design, Fuel Tank Construction, Oxidizer Tank, Oxidizer Tank Design, Oxidizer Tank Construction, Tank Attachment Structure, and Oxidizer Feedline
Structures	Wing Box Technology, Nose, Thrust Structure, Secondary Structures, Wing Shape, Wing Structure, Hot Elevon Structure, Canards, Vertical Tail Location, Vertical Tail Design, and Body Flap
TPS	Aerothermal Performance, TPS Technology Effectiveness, Wing Group TPS, Tail Group TPS, Body Group TPS, and Insulation
Propulsion	Engine Selection, Engine Integration, Pressurant, and RCS
Vehicle-Wide	Budget, Single-Stage Vehicle Performance, and G-limit

system. Finally, two more parameters model alternative chemical systems which would be used for tank pressurant and the RCS system.

The final step in the identification of parameters is assigning each parameter to a subsystem group for later analysis. For this problem four subsystems are being tracked: propellant tanks, vehicle structures, TPS, and propulsion systems. Additionally, certain parameters such as budget and single-stage vehicle performance affect the entire vehicle; these parameters are placed in a separate category for parameters which describe the total vehicle-wide margin. The assignment of parameters to categories is documented in Table 18.

Budget	Baseline Budget	Slight Reduction	Significant Reduction	Slight Increase	
Aerothermal Performance	Baseline Requirements	Reduced Heat Load	Greatly Reduced Heat Load	Increased Heat Load	
Single-Stage Vehicle Performance	Baseline Requirements	Increased Threshold	Decreased Threshold		
Transverse G-Limit	3	4.5	6		
Wing Box	Meets Threshold	Meets Objective	Exceeds Threshold	Does not Meet Threshold	
Composite Tanks	Meets Threshold	Meets Objective	Exceeds Threshold	Does not Meet Threshold	Non-inclusion
Common Bulkhead	Meets Threshold	Meets Objective	Exceeds Threshold	Does not Meet Threshold	Non-inclusion
TPS Technology Effectiveness	Meets Threshold	Meets Objective	Exceeds Threshold	Does not Meet Threshold	
Nose	Baseline	More Volume Necessary			
Fuel Tank	Baseline	Increased Ullage Pressure	Increased Stiffness Req	Both	
Fuel Tank Design	Baseline	Monocoque	Orthogrid		
Fuel Tank Construction	Composite	Aluminum	Aluminum Lithium		
Oxidizer Tank	Baseline	Increased Ullage Pressure	Increased Stiffness Req	Both	
Oxidizer Tank Design	Baseline	Monocoque	Orthogrid		
Oxidizer Tank Construction	Composite	Aluminum	Aluminum Lithium		
Tank Attachment Structure	Common Bulkhead	Intertank			
Thrust Structure	Baseline	Under-estimation of loads	Over-estimation of loads		
Secondary Structures	Baseline	Additional Structure Req	Potential Weight Savings		
Wing Shape	High L/D	Standard Redesign	High L/D Redesign	Add stiffness for flutter	
Wing Structure	Baseline	Under-estimation of loads	Over-estimation of loads		
Hot Elevon Structure	Baseline	Larger Elevon Required			
Canards	No	Yes			
Vertical Tail Location	wing	Mid-Wing	body		
Vertical Tail Design	Baseline	Larger Tail	More Stiffness Required	Both Larger and Stronger	
Body Flap	Baseline	Larger Body Flap			
Wing Group TPS	Baseline	Reduced Tech Effectiveness			
Tail Group TPS	Baseline	Reduced Tech Effectiveness			
Body Group TPS	Baseline	Reduced Tech Effectiveness			
Insulation	Baseline	Reduced Tech Effectiveness			
Engine Selection	Merlin	New Engine Baseline	New Engine Growth		
Engine Integration	Boat Tail	Nacelles			
Oxidizer Feedline	External	Internal, Single Wall	Internal, Vacuum Jacketed		
Pressurant	Helium	GN2			
RCS	MMH/NTO	LOX/Ethanol	New Green Technology		

Figure 78: Morphological Field for FAST RFS F

8.3.2 Identification of Conditions

The next step of EMA is to complete the morphological field by identifying the alternative conditions for each parameter. The alternatives defined in this section represent alternatives studied in design studies which would have been conducted during earlier phases of design or alternatives which would have been examined by systems engineers performing a risk/opportunity analysis in the vehicle program office. The complete morphological field can be seen in Figure 78; because the overall hybrid method assumes a baseline as part of the analysis, the conditions corresponding to the FAST RFS F baseline are highlighted in green. This section details the identification of conditions used to describe the baseline alternatives for the RFS F.

8.3.2.1 *Programmatics*

The programmatic parameters make up the first four parameters in Figure 78. In order to represent political uncertainties over funding issues, the budget has four different options—the baseline budget, a slight increase in budget, and two varying degrees of budget reductions. Two levels of budget reduction representing a slight and significant reduction are included to allow program managers to define discrete levels of risk mitigation in the face of growing budget cuts.

Requirements uncertainty is modeled in the through aerothermal performance, vehicle performance, and g-limit requirements. The conditions defined in this section represent options which were present during the initial requirements exploration stages of the FAST program. The aerothermal performance parameter is given a baseline requirement, a more stringent version of the requirement, and two varying degrees of requirement relaxation. Two degrees of requirements relaxation are offered because the aerothermal requirements will directly drive the material selection of the TPS, a discreet baseline change unable to be modeled by range estimating. Requirements uncertainty is further modeled through single-stage vehicle performance; this requirement is represented by its baseline requirement and increased or decreased performance thresholds. Finally, the g-limit requirement has options of 3, 4.5, and 6 gees; these g-limits considered during the program’s evolution and early

design studies. [22]

8.3.2.2 Technologies

The next four parameters in Figure 78 model the technology development programs. For each technology development program, conditions were assigned to enable each program to meet the program threshold, not meet the threshold, exceed the threshold, or meet the objective. For the case of composite tanks, the option for a failed technology development is included; the selection of this option will make it impossible for the model to select a composite tank construction or common bulkhead in the vehicle physical decomposition.

8.3.2.3 Vehicle Structures

The modeling of the structural subsystems accounts for the next several parameters. This can be further broken down into tankage structures, aerodynamic structures, and secondary/thrust structures. Many of the options discussed in this subsection represent alternatives studied in early design and risk mitigation studies of the FAST program.

Three parameters describe the propellant tanks; both the fuel and oxidizer tanks have the same conditions because they are similar structures. The parameter for the tank describes the loading requirements for the tank. These requirements include the baseline requirements or different increased loading requirements. The construction parameter includes options for construction material. The composite baseline is available along with options for metallic structures. Finally, the design parameters specify the structural design philosophy— the externally stiffened baseline, a monocoque, or an orthogrid. In addition to the parameters which describe the individual tanks, the parameter for the tank attachment structure allows the selection of a common bulkhead or an intertank.

The main wing is modeled by two parameters. The wing structure parameter includes options for the baseline load case along with the over and under-estimation of loads. The wing shape parameter enables options for two different wings— a standard wing and a L/D wing which was baselined into the RFS F. [22, 98] The baseline L/D is highlighted in green; from there options allow redesigns for both the standard wing and high L/D wing. The option to redesign the wing exists because an experimental vehicle like this is known to

have issues with pitch trim stability. [113, 112] Finally, an option exists to add stiffness to account for flutter.

The vertical tail is also modeled using two parameters. The parameter governing the vertical tail location allows the stabilizer to be placed on the wing tip, mid-wing, or on the fuselage. This tail can be redesigned to increase tail size in order to increase directional stability, to add stiffness for flutter, or both.

The control surfaces are modeled using the hot elevon structure, body flap, and canard parameters. Both the body flap and elevon parameters present options for the baseline or larger control surface to improve control power. Similarly, the option to add canards exists because of the possibility of requiring more control power.

The secondary/thrust structures are made up of the nose, trust structure, secondary structure, and control surface parameters. Each of these parameters has a baseline option and options which will trigger a redesign. Both the thrust structure and secondary structures have options which would trigger a mass gain or a mass savings while the nose only has an option which will trigger mass growth.

8.3.2.4 Vehicle TPS

Each parameter modeling the TPS has identical conditions. Each has a condition representing the baseline technology level and a condition representing reduced technology effectiveness. These conditions are meant to work with the aerothermal requirement parameter and TPS technology effectiveness parameter through relationships.

8.3.2.5 Vehicle Propulsion

Propulsion is modeled by five parameters which reflect propulsion studies carried out during the development of the RFS F. The engine selection parameter allows the choice between the existing new engine design, an off-the-shelf SpaceX Merlin engine, or a new engine with additional mass growth; this option for additional mass growth on top of the mass growth predicted by range estimated was added to account for the fact that it is a new engine. The engines can be mounted on the aft fuselage as part of a tradition boat tail or on nacelles; the nacelle concept was incorporated as a way to improve the maintainability of the propulsion

system. [35] The oxidizer feedline can be an simple external feedline, or two different designs for feedline which runs through the fuel tank. Finally, pressurants can be either helium or nitrogen while the RCS system can be either hydrazine based, LOX/ethanol, or a new green RCS technology.

8.3.3 Identification of Condition Attributes

The next step in EMA is the identification of condition attributes. For this problem, each condition will have a likelihood, multiplicative effect, and additive effect. As with the Orbiter example problem in Section 7.2.2.3, the probabilities are estimated by the author with a bias on probability values placed on baseline conditions. Additionally, while each condition has potential values for both additive and multiplicative effects, some conditions will not have a first order effect— a multiplicative effect of 1 and an additive effect of 0 do not affect the modeled weight. Instead, these conditions will trigger relationships which will impart a second order effect on the modeled weight. Finally, baseline elements highlighted in green have no impact on the modeled weight compared to the original WBS in Table 16.

The programmatics matrix can be seen in Figure 79. The first item covered in this matrix is the budget outlay for the program. A reduction in budget would affect the overall weight of the vehicle because it would hinder a mass control program; similarly, a budget increase can enhance a mass control program resulting in a reduction in weight. This first order effect is modeled through multiplicative relationships. Similarly, the single-stage vehicle performance has a first order effect on the vehicle as it represents the overall required performance; a decreased requirement threshold will lead to less required margin while an increased threshold will require additional margin. Finally, both the requirements on aerothermal performance and transverse G-limit have no first order effects on the vehicle weight. The conditions in these parameters affect the weight of the vehicle through relationships.

The matrix describing the technology development programs can be seen in Figure 80. None of the technology development programs listed in this matrix have first order effects. All technology development programs will affect the individual parameters in the vehicle

Budget	Baseline Budget	Slight Reduction	Significant Reduction	Slight Increase
Probability	0.4	0.25	0.2	0.15
Multiplicative Effect	1	1.01	1.03	0.99
Additive Effect	0	0	0	0
Aerothermal Performance	Baseline Requirements	Reduced Heat Load	Greatly Reduced Heat Load	Increased Heat Load
Probability	0.75	0.1	0.05	0.1
Multiplicative Effect	1	1	1	1
Additive Effect	0	0	0	0
Single-Stage Vehicle Performance	Baseline Requirements	Increased Threshold	Decreased Threshold	
Probability	0.6	0.1	0.3	
Multiplicative Effect	1	1.1	0.9	
Additive Effect	0	0	0	
Transverse G-Limit	3	4.5	6	
Probability	0.15	0.75	0.1	
Multiplicative Effect	1	1	1	
Additive Effect	0	0	0	

Figure 79: Programmatic Matrix

physical decomposition through relationships.

The matrix describing the structural subsystems can be seen in Figure 81. As each of these parameters only describe subsystems, the multiplicative effect is unity. The additive effect for baseline conditions mirrors the baseline weight from Table 16. The decision to model baseline conditions' additive effect as the original weight estimation allows the full subsystem weight to subsequently be modified via multiplicative relationships. For example, the common bulkhead structural weight can be multiplied by a factor depending on which common bulkhead technology development program condition is chosen. In addition to modification through relationships, each parameter contains alternative baseline configuration conditions whose additive effects reflect a percentage offset from the original weight.

In this decomposition, there are several structural subsystems described by multiple parameters. Specifically, both propellant tanks, the wing, and vertical tail are modeled using multiple parameters. In each of these cases, only one of the parameters carries the full weight allocation for the subsystem from Table 16. In order to avoid double-counting weight changes, the other parameters have no first order effect on the vehicle weight; instead, these additional parameters will affect the primary parameter's additive effect through relationships.

Wing Box Structures	Meets Threshold	Meets Objective	Exceeds Threshold	Does not Meet Threshold	
Probability	0.5	0.1	0.2	0.2	
Multiplicative Effect	1	1	1	1	
Additive Effect	0	0	0	0	
Composite Tanks	Meets Threshold	Meets Objective	Exceeds Threshold	Does not Meet Threshold	Non-inclusion
Probability	0.5	0.1	0.15	0.1	0.15
Multiplicative Effect	1	1	1	1	1
Additive Effect	0	0	0	0	0
Common Bulkhead	Meets Threshold	Meets Objective	Exceeds Threshold	Does not Meet Threshold	Non-inclusion
Probability	0.5	0.1	0.15	0.1	0.15
Multiplicative Effect	1	1	1	1	1
Additive Effect	0	0	0	0	0
TPS Technology Effectiveness	Meets Threshold	Meets Objective	Exceeds Threshold	Does not Meet Threshold	
Probability	0.5	0.1	0.2	0.2	
Multiplicative Effect	1	1	1	1	
Additive Effect	0	0	0	0	

Figure 80: Technology Development Matrix

The matrix describing the TPS conditions can be seen in Figure 82. Each baseline condition has a probability of 60%, and each reduced technology effectiveness condition has a probability of 40%. The additive effect of each baseline condition reflects the original estimated weight of each TPS subsystem in Table 16. The additive effect for each parameter's reduced technology effectiveness conditions is a 5% increase in weight over the baseline conditions. These condition attributes are intended to be modified by relationships during the execution of the model; specifically, the technology development program parameters will modify these parameters as necessary.

The final set of parameters, the propulsion elements, can be seen in Figure 83. Unlike the structures and TPS portions of the overall matrix, most propulsion elements are only represented by their delta from the original baseline. This change is made for propulsion elements because there are few relationships from other parts of the model which will need to modify the full weight of these subsystems. Additionally, the engine integration, oxidizer feedline, and RCS subsystems' mass impacts are assumed representative weights to complete the model. The weight impact of the pressurant is based on the weight savings which could be accomplished by using helium instead of nitrogen. [143]

The final step as part of the identification of effects is determining the value of the baseline weight used in Equation 26. Because many of the baseline conditions contain

Nose	Baseline	More Volume Necessary		
Probability	0.75	0.25		
Multiplicative Effect	1	1		
Additive Effect	442	486.2		
Fuel Tank	Baseline	Increased Ullage Pressure	Increased Stiffness Req	Both
Probability	0.6	0.15	0.15	0.1
Multiplicative Effect	1	1	1	1
Additive Effect	662	675.24	695.1	708.34
Fuel Tank Design	Baseline	Monocoque	Orthogrid	
Probability	0.5	0.25	0.25	
Multiplicative Effect	1	1	1	
Additive Effect	0	0	0	
Fuel Tank Construction	Composite	Aluminum	Aluminum Lithium	
Probability	0.6	0.25	0.15	
Multiplicative Effect	1	1	1	
Additive Effect	0	0	0	
Oxidizer Tank	Baseline	Increased Ullage Pressure	Increased Stiffness Req	Both
Probability	0.6	0.15	0.15	0.1
Multiplicative Effect	1	1	1	1
Additive Effect	867	884.34	910.35	927.69
Oxidizer Tank Design	Baseline	Monocoque	Orthogrid	
Probability	0.75	0.25	0	
Multiplicative Effect	1	1	1	
Additive Effect	0	0	0	
Oxidizer Tank Construction	Composite	Aluminum	Aluminum Lithium	
Probability	0.6	0.25	0.15	
Multiplicative Effect	1	1	1	
Additive Effect	0	0	0	
Tank Attachment Structure	Common Bulkhead	Intertank		
Probability	0.75	0.25		
Multiplicative Effect	1	1		
Additive Effect	109	327		
Thrust Structure	Baseline	Under-estimation of loads	Over-estimation of loads	
Probability	0.7	0.2	0.1	
Multiplicative Effect	1	1	1	
Additive Effect	258	270.9	245.1	
Secondary Structures	Baseline	Additional Structure Req	Potential Weight Savings	
Probability	0.6	0.25	0.15	
Multiplicative Effect	1	1	1	
Additive Effect	681	715.05	646.95	
Wing Shape	High L/D	Standard Redesign	High L/D Redesign	Add stiffness for flutter
Probability	0.5	0.1	0.2	0.2
Multiplicative Effect	1	1	1	1
Additive Effect	0	0	0	0
Wing Structure	Baseline	Under-estimation of loads	Over-estimation of loads	
Probability	0.6	0.3	0.1	
Multiplicative Effect	1	1	1	
Additive Effect	3240	3402	3078	
Hot Elevon Structure	Baseline	Redesign		
Probability	0.8	0.2		
Multiplicative Effect	1	1		
Additive Effect	209	219.45		
Canards	No	Yes		
Probability	0.8	0.2		
Multiplicative Effect	1	1		
Additive Effect	0	324		
Vertical Tail Location	wing	Mid-Wing	body	
Probability	0.7	0.15	0.15	
Multiplicative Effect	1	1	1	
Additive Effect	0	0	0	
Vertical Tail Design	Baseline	Larger Tail	More Stiffness Required	Both Larger and Stronger
Probability	0.5	0.2	0.15	0.15
Multiplicative Effect	1	1	1	1
Additive Effect	285	313.5	299.25	327.75
Body Flap	Baseline	Larger Body Flap		
Probability	0.75	0.25		
Multiplicative Effect	1	1		
Additive Effect	225	236.25		

Figure 81: Structural Subsystem Matrix

Wing Group TPS	Baseline	Reduced Tech Effectiveness
Probability	0.6	0.4
Multiplicative Effect	1	1
Additive Effect	276.2	290.01
Tail Group TPS	Baseline	Reduced Tech Effectiveness
Probability	0.6	0.4
Multiplicative Effect	1	1
Additive Effect	67.1	70.455
Body Group TPS	Baseline	Reduced Tech Effectiveness
Probability	0.6	0.4
Multiplicative Effect	1	1
Additive Effect	714.5	750.225
Insulation	Baseline	Reduced Tech Effectiveness
Probability	0.6	0.4
Multiplicative Effect	1	1
Additive Effect	79	82.95

Figure 82: TPS Matrix

Engine Selection	Merlin 1-C	New Engine Baseline	New Engine Growth
Probability	0.2	0.6	0.2
Multiplicative Effect	1	1	1
Additive Effect	1145	1500	1650
Engine Integration	Boat Tail	Nacelles	
Probability	0.25	0.75	
Multiplicative Effect	1	1	
Additive Effect	-75	0	
Oxidizer Feedline	External	Internal, Single Wall	Internal, Vacuum Jacketed
Probability	0.25	0.25	0.5
Multiplicative Effect	1	1	1
Additive Effect	50	25	0
Pressurant	Helium	GN2	
Probability	0.25	0.75	
Multiplicative Effect	1	1	
Additive Effect	-355	0	
RCS	MMH/NTO	LOX/Ethanol	New Green Technology
Probability	0.15	0.75	0.1
Multiplicative Effect	1	1	1
Additive Effect	50	0	50

Figure 83: Propulsion Matrix

weight estimates, the baseline weight used for Equation 26 is not equal to the dry weight listed in Table 16. For this problem, the number used is 7,465 lbs— the combined mass of all subsystem weight estimates not accounted for by matrix parameters.

8.3.4 Identification of Relationships

8.3.4.1 Organization of Parameters

The first part of the relationship identification phase is to define necessary sub-matrices in order to minimize one-way relationships. For this problem, three sub-matrices are used: programmatic, technology development, and vehicle decomposition. These three matrices create a hierarchy of one-way relationships. One-way relationships from the programmatic matrix can affect the technology development programs and the vehicle, and one-way relationships from the technology development matrix can only affect the vehicle matrix.

The programmatic matrix can be seen in Figure 79 and contains the budget and requirements parameters. Similarly, the technology development matrix is seen in Figure 80. The vehicle matrix is a concatenation of the matrices contained in Figures 81, 82, and 83.

8.3.4.2 Identifying One-Way Relationships

The one-way probability relationships between different funding profiles and the technology development programs can be seen in Figure 84; this figure omits the four relationships which could affect multiplicative and additive effects because these entries are trivial. Smaller funding profiles lead to less effective technology development programs. Similarly, a budget increase will increase the chances of technologies meeting or exceeding their thresholds. The one-way probability relationships between the different funding profiles and alternative vehicle baseline options can be seen in Figure 85; similarly, this figure omits the four relationships which could affect multiplicative and additive effects because these entries are trivial. These relationships model the idea that budget reductions will lead to more conventional design decisions; for example, a smaller budget could lead to choosing aluminum tanks and an intertank instead of composite tanks with a common bulkhead.

The one-way relationships between requirements parameters and vehicle options can

	Wing Box				Composite Tanks					Common Bulkhead					TPS Technology Effect				
	Meets Threshold	Meets Objective	Exceeds Threshold	Does not Meet Threshold	Meets Threshold	Meets Objective	Exceeds Threshold	Does not Meet Threshold	Non-inclusion	Meets Threshold	Meets Objective	Exceeds Threshold	Does not Meet Threshold	Non-inclusion	Meets Threshold	Meets Objective	Exceeds Threshold	Does not Meet Threshold	
Baseline Budget	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Slight Reduction	0.9	0.5	0.8	1.5	0.9	0.5	0.75	1.5	1.5	0.9	0.5	0.75	1.5	1.5	0.9	0.5	0.75	1.5	
Significant Reduction	0.75	0.25	0.5	2	0.75	0.3	0.5	2	2	0.75	0.25	0.5	2	2	0.75	0.25	0.5	2	
Slight Increase	1.25	1.5	2	0.75	1.25	1.5	2	0.75	0.5	1.25	1.5	2	0.75	0.5	1.25	1.5	2	0.75	

Figure 84: Probability Relationships Between Budget Alternatives and Technology Development Projects

	Fuel Tank C			Oxidizer Tank C			Tank Attac		Engine Selection			Oxidizer Feed		
	Composite	Aluminum	Aluminum Lithium	Composite	Aluminum	Aluminum Lithium	Intertank	Common Bulkhead	Merlin	New Engine Baseline	New Engine Growth	External	Internal, Single Wall	Internal, Vacuum Jacketed
Baseline Budget	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Slight Reduction	0.85	1.5	1.1	0.85	1.5	1.1	1.5	0.85	2	0.75	1.25	2	0.75	0.5
Significant Reduction	0.5	2	1.3	0.5	2	1.3	3	0.75	3	0.5	0.75	3	0.5	0.3
Slight Increase	2	0.85	1	1.5	0.85	1	0.85	2	0.85	2	0.85	0.85	0.9	1.2

Figure 85: Probability Relationships Between Budget Alternatives and Vehicle Subsystems

	Wing Group TPS		Tail Group TPS		Body Group TPS		Insulation	
	Baseline	Reduced Tech	Baseline	Reduced Tech	Baseline	Reduced Tech	Baseline	Reduced Tech
	Baseline Requirements	1	1	1	1	1	1	1
Reduced Heat Load	0.75	0.8	0.8	0.8	0.75	0.8	1	1
Greatly Reduced Heat	0.5	0.6	0.5	0.6	0.5	0.6	1	1
Increased Heat Load	1.25	1.3	1.3	1.3	1.25	1.3	1	1

Figure 86: Additive Effect, Multiplicative Relationships Between Aerothermal Requirements and Vehicle TPS

be seen in Figures 86 and 87. Figure 86 shows multiplicative relationships with the additive effects of the TPS group parameters. Increased aerothermal requirements lead to additional TPS weight and vice-versa. All other relationships between aerothermal requirements and TPS vehicle options are trivial. Similarly, higher G-limit requirements drive increased weight allocations for load bearing parts of the vehicle structure. Figure 87 shows that probability relationships and multiplicative relationships affecting additive effects exist between G-limit requirements and vehicle structural subsystems; an increase in g-limit requirements increases the chances of a redesign for increased tank stiffness. The additive relationship for the additive effect and relationships affecting the multiplicative effect of structural conditions are omitted from Figure 87 because they contain trivial 'no effect' entries.

The one-way relationships between technology development programs and vehicle options can be seen in Figures 88, 89, 90, and 91; these figures omit relationships which do not affect vehicle options. These technologies affect vehicle weights through multiplicative relationships. A more successful technology development program will result in lower weights and vice-versa. Additionally, probability relationships exist which could trigger design changes to less mass efficient design options if technology development programs progress poorly. These probability relationships implement a central design feature of the model— design options and technology development programs are modeled separately. This

	Fuel Tank				Oxidizer Tank				Tank Att	Wing Structure		Canards			
	Baseline	Increased Ullage Pressure	Increased Stiffness Req	Both	Baseline	Increased Ullage Pressure	Increased Stiffness Req	Both	Intertank	Common Bulkhead	Baseline	Under-estimation of loads	Over-estimation of loads	No	Yes
3															
Probability	1	1	0.5	0.5	1	1	0.5	0.5	1	1	1	1	1	1	1
Add Eff, Mult Rel	0.85	0.85	0.9	0.9	0.85	0.9	0.85	0.85	0.85	0.95	0.85	0.85	0.85	1	0.9
4.5															
Probability	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Add Eff, Mult Rel	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
6															
Probability	1	1	2	2	1	1	2	2	1	1	1	1	1	1	1
Add Eff, Mult Rel	1.25	1.25	1.3	1.3	1.25	1.3	1.25	1.25	1.25	1.05	1.25	1.25	1.25	1	1.3

Figure 87: Relationships Between G-limit Requirements and Vehicle Substructures

decision is intended to note the difference between technology development and integration; a technology development program may develop technologies which are impractical, expensive, or fruitless to include on a vehicle.

8.3.4.3 Identifying Two-Way Relationships

The programmatic relationship matrix can be seen in Figure 92. This figure only contains probability relationships because there are no relationships affecting multiplicative or additive effects. The major relationships in this matrix stem from interactions between the budget outlays and requirements. In this model, if the budget is reduced, then it is more likely that requirements will be relaxed. Similarly, if the budget is increased, it is more likely that additional aerothermal and vehicle performance requirements will be levied.

There are no relationships between members of the technology development matrix. There can be no relationships affecting non-existent effects because none of the conditions have a multiplicative or additive effect on the vehicle mass. Furthermore, there are no probability relationships because each technology development program is assumed to be independent.

	Wing Structure		
	Baseline	Under-estimation of loads	Over-estimation of loads
Meets Threshold	1	1	1
Meets Objective	0.85	0.85	0.9
Exceeds Threshold	0.95	0.95	1
Does not Meet Threshold	1.1	1.1	1.1

Figure 88: Additive Effect, Multiplicative Relationships Between Wing Box Technology Development and the Wing Structure

	Fuel Tank			Oxidizer Tank		
	Composite	Aluminum	Aluminum Lithium	Composite	Aluminum	Aluminum Lithium
Meets Threshold						
Probability	1	1	1	1	1	1
Add Eff, Mult Rel	1	1	1	1	1	1
Meets Objective						
Probability	1.2	1	1	1.2	1	1
Add Eff, Mult Rel	0.9	1	1	0.9	1	1
Exceeds Threshold						
Probability	1.1	1	1	1.1	1	1
Add Eff, Mult Rel	0.95	1	1	0.95	1	1
Does not Meet Threshold						
Probability	0.75	1	1	0.75	1	1
Add Eff, Mult Rel	1.1	1	1	1.1	1	1
Non-Inclusion						
Probability	0	1	1	0	1	1
Add Eff, Mult Rel	1	1	1	1	1	1

Figure 89: Relationships Between Composite Tank Technology Development and the Propellant Tanks

	Tank Att	
	Intertank	Common Bulkhead
Meets Threshold Probability	1	1
Add Eff, Mult Rel	0	0
Meets Objective Probability	1	1.5
Add Eff, Mult Rel	0	0.9
Exceeds Threshold Probability	1	1.25
Add Eff, Mult Rel	0	0.95
Does not Meet Threshold Probability	1	0.75
Add Eff, Mult Rel	0	1.1
Non-Inclusion Probability	1	0
Add Eff, Mult Rel	0	0

Figure 90: Relationships Between Common Bulkhead Technology Development and Vehicle Structures

	Wing TPS		Tail TPS		Body TPS		Insulation	
	Baseline	Reduced Tech	Baseline	Reduced Tech	Baseline	Reduced Tech	Baseline	Reduced Tech
Meets Threshold Probability	1	1	1	1	1	1	1	1
Add Eff, Mult Rel	1	1	1	1	1	1	1	1
Meets Objective Probability	1	0.5	1	0.5	1	0.5	1	0.5
Add Eff, Mult Rel	0.75	1	0.8	1	0.75	1	0.75	1
Exceeds Threshold Probability	1	0.75	1	0.8	1	0.8	1	0.75
Add Eff, Mult Rel	0.9	1	0.9	1	0.9	1	0.9	1
Does not Meet Threshold Probability	0.75	1.5	0.8	1.5	0.75	1.5	0.75	1.5
Add Eff, Mult Rel	1.5	2	1.5	2	1.5	2	1.5	2

Figure 91: Relationships Between TPS Technology Development and Vehicle TPS

		Budget				Aerothermal				Vehicle		
		Baseline Budget	Slight Reduction	Significant Reduction	Slight Increase	Baseline Requirements	Reduced Heat Load	Greatly Reduced Heat Load	Increased Heat Load	Baseline Requirements	Increased Threshold	Decreased Threshold
Aero-thermal	Baseline Requirements	1	1	1	1							
	Reduced Heat Load	1.5	2	3	1							
	Greatly Reduced Heat Load	1	1.5	2	1							
	Increased Heat Load	1.2	0.8	0.5	1.3							
Vehicle	Baseline Requirements	1	1	1	1	1	1	1	1			
	Increased Threshold	1	0.9	0.9	1.5	1	1	1	1			
	Decreased Threshold	1	1.5	2	1	1	1	1	1			
G-Limit	3	1	1.3	1.5	1	1	1	1	1	1	1	1
	4.5	1	1	1	1	1	1	1	1	1	1	1
	6	1	1	1	1	1	1	1	1	1	1	1

Figure 92: Programmatics Relationship Matrix

The vehicle relationship matrix is very large (69x68) and cannot be fully reproduced in a single figure. Furthermore, the matrix is very sparse. This matrix only contains probability and multiplicative relationships for additive effects; all other relationships are trivial and are omitted from these figures. Furthermore, these figures will omit many of the 'non effect' entries in matrices to improve the readability of larger figures. The remainder of this section documents the portions of the relationship matrix which has interesting probability and multiplicative relationships.

The relationship matrix in Figure 93 highlights a portion of the probability relationships in the vehicle matrix. This sub-matrix highlights the relationships between tank construction and design as well as correlations between the oxidizer and fuel tanks. The first thing to highlight in this matrix is that composite construction and orthogrid design are incompatible. The next trend is monocoque and orthogrid designs are more likely to occur with aluminum and aluminum lithium construction. Finally, there is a link between both the oxidizer and fuel tank sections– if one material or design is chosen for a single tank, then

		Fuel Tank				F Tank Des			F Tank Con			Oxidizer Tank				O Tank Des		
		Baseline	Increased Ullage Pressure	Increased Stiffness Req	Both	Baseline	Monocoque	Orthogrid	Composite	Aluminum	Aluminum Lithium	Baseline	Increased Ullage Pressure	Increased Stiffness Req	Both	Baseline	Monocoque	Orthogrid
Fuel Tank Design	Baseline																	
	Monocoque																	
	Orthogrid																	
Fuel Tank Construct	Composite										0							
	Aluminum								1.5	1.5								
	Aluminum Lithium								1.5	1.5								
Oxidizer Tank Design	Baseline																	
	Increased Ullage Pressure																	
	Increased Stiffness Req			2														
	Both				2													
Oxidizer Tank Construct	Baseline																	
	Monocoque																	
	Orthogrid																	
	Composite								1.5									0
	Aluminum								1.5								1.5	1.5
	Aluminum Lithium									1.5							1.5	1.5

Figure 93: Tank Structures Probability Relationship Matrix

		Wing TPS		Tail TPS		Body TPS	
		Baseline	Reduced Tech Effectiveness	Baseline	Reduced Tech Effectiveness	Baseline	Reduced Tech Effectiveness
Tail TPS	Baseline	1.5					
	Reduced Tech Effectiveness		1.5				
Body TPS	Baseline			1.5			
	Reduced Tech Effectiveness				1.5		
Insul	Baseline					1.5	
	Reduced Tech Effectiveness						1.5

Figure 94: TPS Probability Relationship Matrix

the other tank is more likely to have a similar design and construction.

Figure 94 shows probability relationships between TPS choices. This figure illustrates the correlations in probability between TPS choices. If a baseline condition is chosen for one TPS parameter, then it is more likely that additional baseline conditions will be chosen within the TPS group. Similarly, if one ends on reduced technology effectiveness, then the other TPS parameters will also have a higher chance of a reduced technology effectiveness.

In addition to the probability relationships between tank structures and TPS, there are sparse probability relationships which model correlations due to the loading of the structure. Probability relationships exist between the wing structures, secondary structures, and the thrust structure. If the a structures loads are chosen to be under-estimated, then it is likely that the other parameters' selection will end up as under-estimated. Similarly, an over-estimation of loads will result in other over-estimations to account for the new loading. Another probability relationship exists between the tail design and the wing shape. A stronger tail will in turn require a stronger wing to accommodate the additional wing load.

The multiplicative relationships between additive effects in the vehicle matrix can be

		Fuel Tank				Oxidizer Tank			
		Baseline	Increased Ullage Pressure	Increased Stiffness Req	Both	Baseline	Increased Ullage Pressure	Increased Stiffness Req	Both
Fuel Tank Design	Baseline	1	1	1	1				
	Monocoque	1.2	1.2	1.2	1.2				
	Orthogrid	1.1	1.1	1.1	1.1				
Fuel Tank Construct	Composite	1	1	1	1				
	Aluminum	1.1	1.1	1.1	1.1				
	Aluminum Lithium	1.1	1.1	1.1	1.1				
Oxidizer Tank Design	Baseline					1	1	1	1
	Monocoque					1.2	1.2	1.2	1.2
	Orthogrid					1.1	1.1	1.1	1.1
Oxidizer Tank Construct	Composite					1	1	1	1
	Aluminum					1.1	1.1	1.1	1.1
	Aluminum Lithium					1.1	1.1	1.1	1.1

Figure 95: Tank Additive Effect, Multiplicative Relationship Matrix

		Wing Shape				Wing Str			Elevon		Canard		Tail Loc			Tail Design			
		High L/D	Standard Redesign	High L/D Redesign	Add stiffness for flutter	Baseline	Under-estimation of loads	Over-estimation of loads	Baseline	Redesign	No	Yes	wing	Mid-Wing	body	Baseline	Larger Tail	More Stiffness Required	Both Larger and Stronger
Wing Str	Baseline	1	0.9	1.1	1.1														
	Under-estimation of loads	1	0.9	1.1	1.1														
	Over-estimation of loads	1	0.9	1.1	1.1														
Elevon	Baseline																		
	Redesign																		
Canard	No																		
	Yes																		
Tail Loc	wing					1	1	1											
	Mid-Wing					1	1	1											
	body					0.9	0.9	0.9											
Tail Design	Baseline					1	1	1					1	1	0.9				
	Larger Tail					1.1	1.1	1.1					1	1	0.9				
	More Stiffness Required					1.1	1.1	1.1					1	1	0.9				
	Both Larger and Stronger					1.2	1.2	1.2					1	1	0.9				
Body Flap	Baseline																		
	Larger Body Flap																		
Wing TPS	Baseline	1	0.9	1.1							1	1.1							
	Reduced Tech Effectiveness	1	0.9	1.1							1	1.1							
Tail TPS	Baseline												1	0.9	0.8	1	1.1	1	1.1
	Reduced Tech Effectiveness												1	0.9	0.8	1	1.1	1	1.1

Figure 96: Wing Additive Effect, Multiplicative Relationship Matrix

seen in Figure 95 and 96. Figure 95 shows how different choices in the design and construction of propellant tanks affects the structural weight of the tank. Figure 96 shows the interaction between tail design choices, wing design choices, control surfaces, and TPS weights.

8.3.5 Analysis

The analysis of this EMA model begins by running a 10,000 case Monte Carlo simulation. While Section 5.4.4.4 shows that a limited amount of Monte Carlo cases are necessary to create a defensible probabilistic analysis, 10,000 cases were chosen in order to carry out analyses described in Section 6.2.4 which require addition cases. This section carries out the three types of analysis detailed in Section 6.2.4: probabilistic analysis of prospective

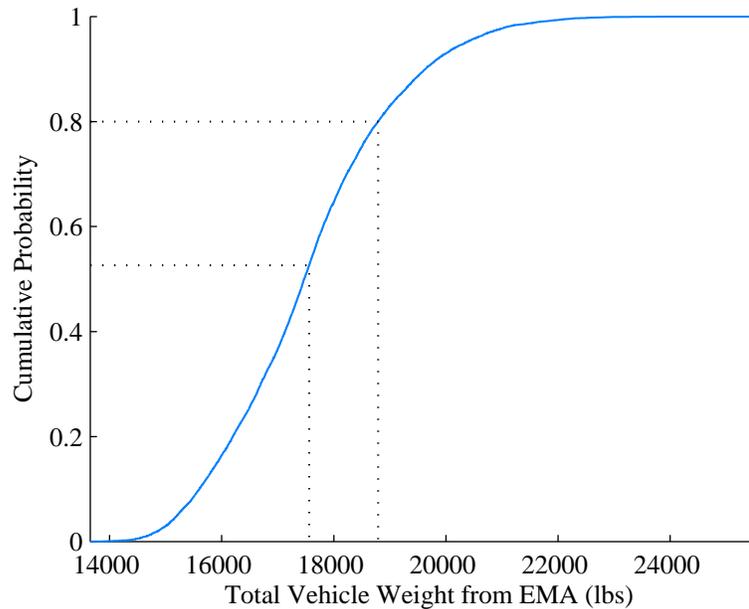


Figure 97: Monte Carlo Results for Vehicle Weight

weights, sensitivity analysis, and what-if analysis.

8.3.5.1 Probabilistic Analysis

The CDF reflecting the results of the 10,000 case Monte Carlo simulation can be seen in Figure 97. The results of the model run provide total weight estimates of the RFS based on potential baseline configuration changes. As with the previous CDFs, both the expected value and 80% quantile value are illustrated. This model produces an expected value weight of 17,561 lbs and an 80% quantile value of 18,790 lbs.

In order to get margins out of this data, the original 17,080 lb baseline dry weight must be subtracted. This results in a CDF which can be seen in Figure 98. This figure highlights the expected value margin of 481 lbs, and 80% quantile value of 1,711 lbs.

While top-line weight and margin estimations are useful, many decision makers would prefer to specify subsystem-level margins which would then add up to a total vehicle margin. Because the parameters of the EMA model were labeled as parts of subsystems, the Monte Carlo simulation can provide information about subsystem margin estimates. The expected value and 80% quantile value for each subsystem identified in Section 8.3.1 can be seen in

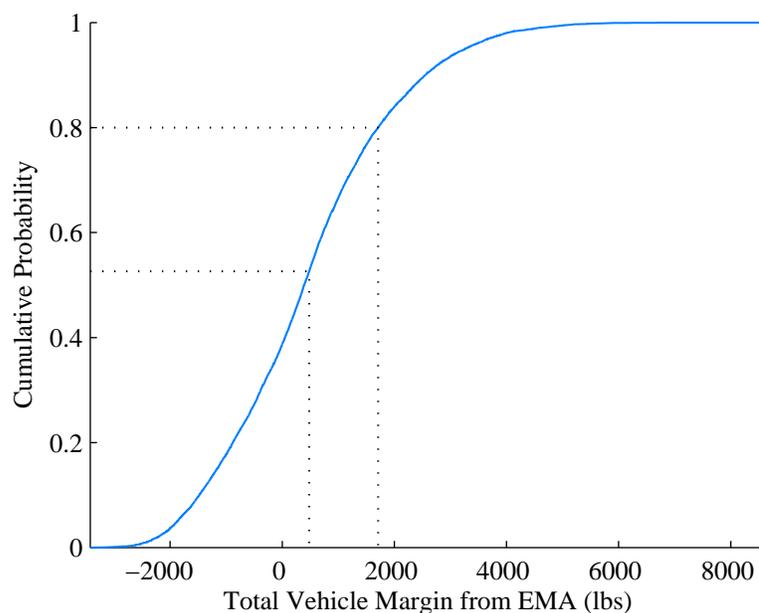


Figure 98: Monte Carlo Results for Vehicle Margin

Table 19: Subsystem Margin Estimation from EMA

Subsystem	Expected Value (lbs)	80% Quantile (lbs)
Propellant Tanks	368	561
Structures	383	876
TPS	244	318
Propulsion	-212	2
Vehicle-Wide	-303	471
Sum	481	2,230

Table 19, and the CDF for each subsystem can be seen in Figure 99 through 103. If each subsystem is allocated via the 80% quantile criterion, then this method results in a more conservative margin estimate by 518 lbs.

These two different methods of looking at margin can have different results because the weight estimates are correlated. Figure 104 shows a scatterplot matrix where each subsystem margin and total margin is plotted against each other. In this matrix, the correlations between estimates can be visualized; additionally, the correlation matrix can be seen in Table 20. From this information, the fact that the structures and propellant tank

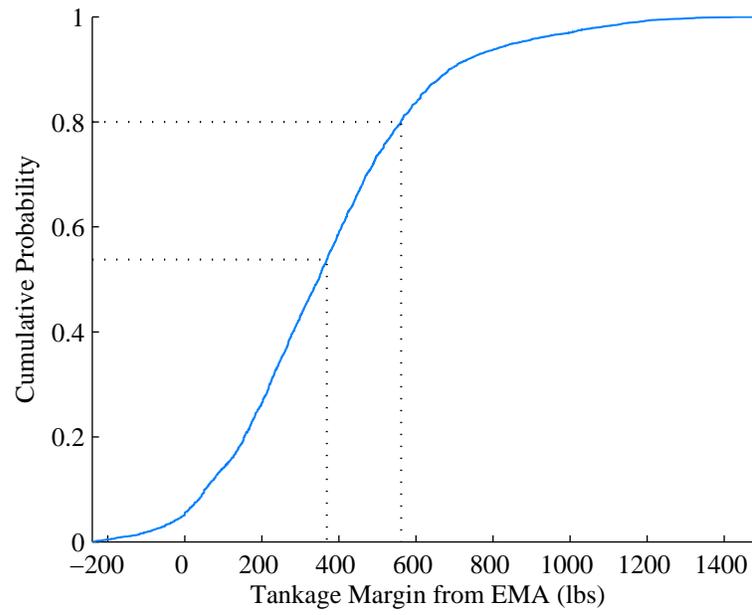


Figure 99: Monte Carlo Results for Tank Structures Margin

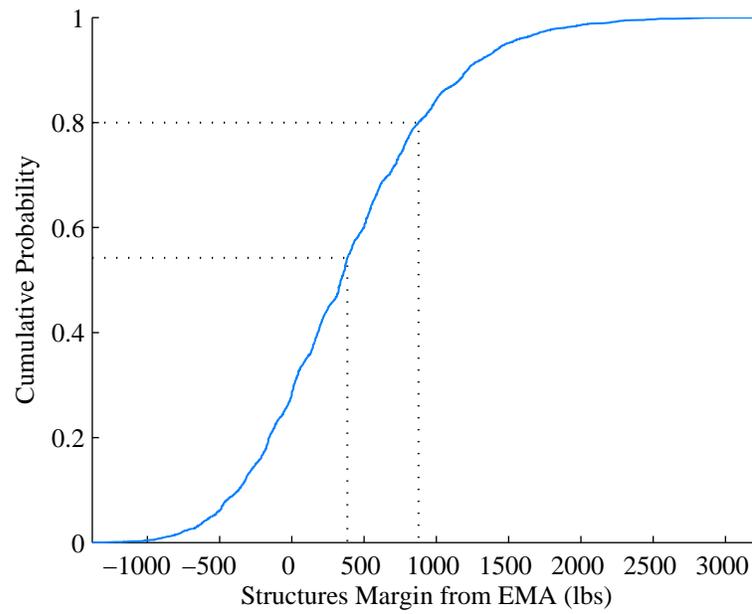


Figure 100: Monte Carlo Results for Structures Margin

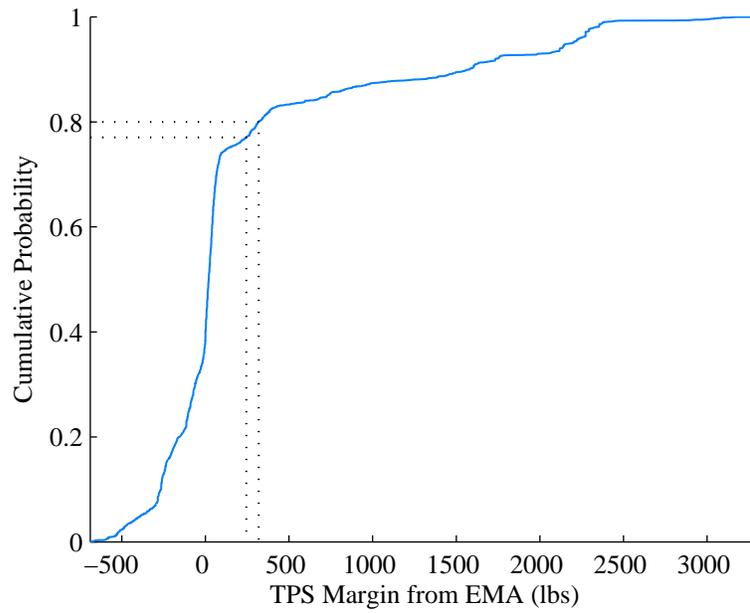


Figure 101: Monte Carlo Results for TPS Margin

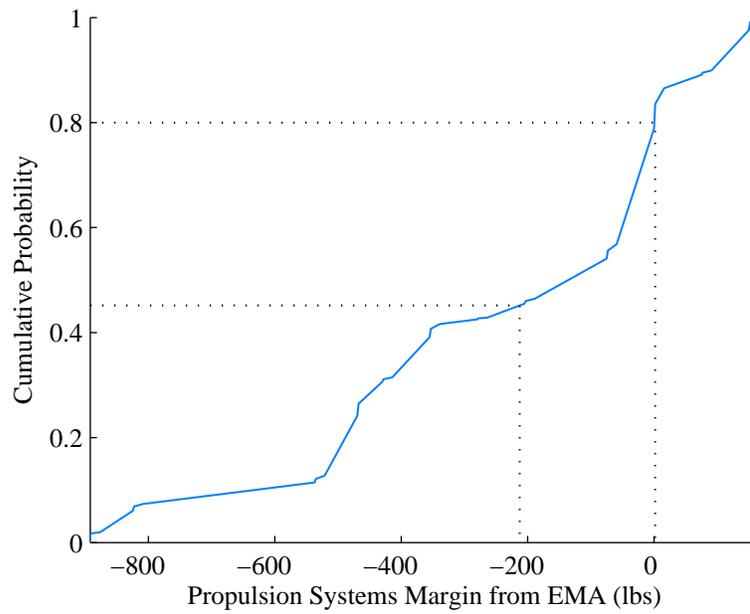


Figure 102: Monte Carlo Results for Propulsion Margin

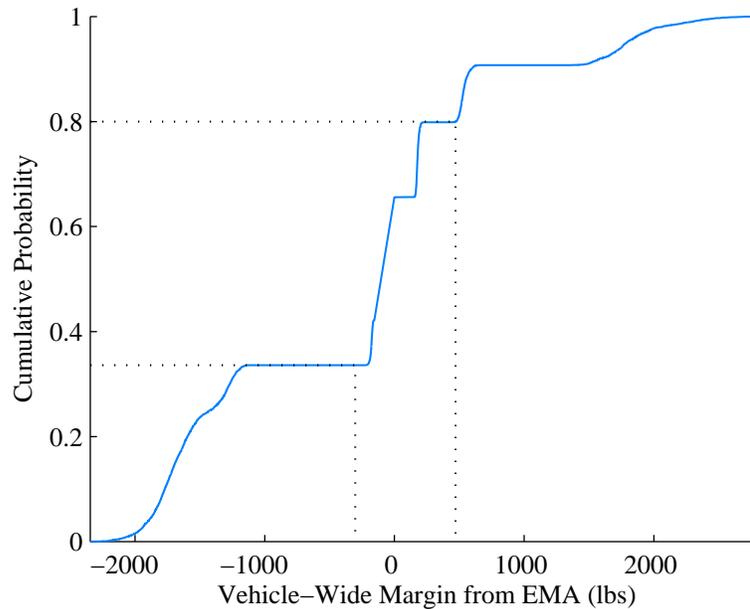


Figure 103: Monte Carlo Results for Vehicle-Wide Margin

weights are correlated is apparent.

These correlations can help decision makers determine how much importance is placed on subsystem estimates compared to total weight estimates. Figure 104 illustrates a filtered Monte Carlo of different estimation priorities. The green dots illustrate cases with sufficient margin when using the 80% value based on the total vehicle margin (Figure 98). Blue dots represent cases which have sufficient margin when both the propellant tank and structures margins are allocated using the subsystem 80% value—the structures and tankage are chosen because these two margins are correlated. Finally, purple dots represent cases which pass both total margin and subsystem margin criterion, and black dots fail all criterion. The different criterion are readily apparent on this plot—the total margin criterion is visualized through a straight line through the bottom and right-hand rows of the matrix while the subsystem criterion cuts out the lower left-hand corner of the propellant tank margin versus structures margin plot. Because purple dots meet both criteria, any remaining green dots represent vehicles which exceed subsystem margins while remaining blue dots exceed total vehicle margin.

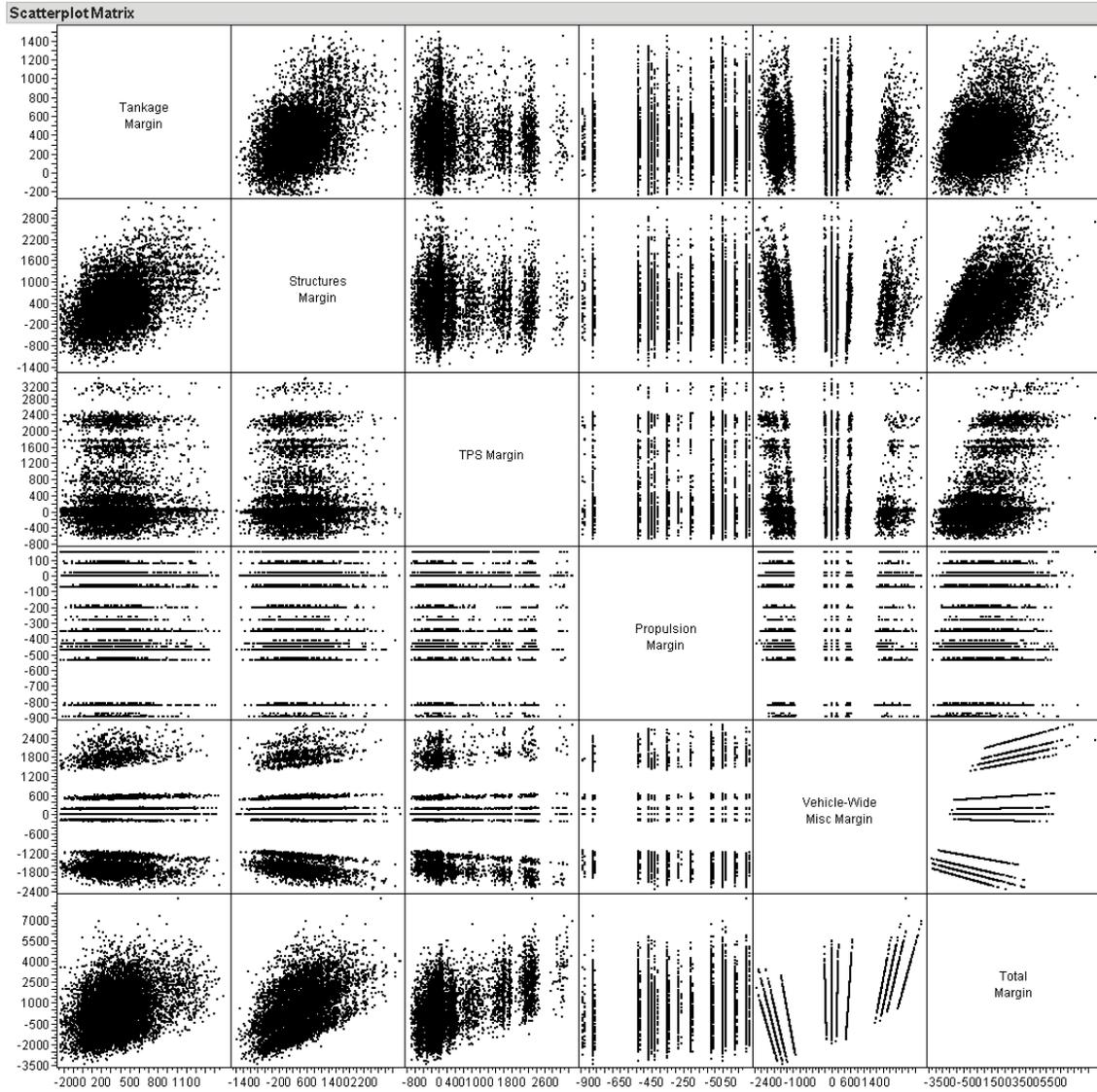


Figure 104: Scatterplot for Subsystem Margins

Table 20: Subsystem Margin Correlations

	Tanks	Structures	TPS	Propulsion	Vehicle-Wide	Total
Propellant Tanks	1.0000	0.3952	0.0145	-0.0336	-0.0051	0.3295
Structures	0.3952	1.0000	0.0259	0.0170	-0.0038	0.4892
TPS	0.0145	0.259	1.0000	-0.0123	0.0010	0.4767
Propulsion	-0.0336	0.0170	-0.0123	1.0000	-0.0236	0.1623
Vehicle-Wide	-0.0051	-0.0038	0.0010	-0.0236	1.0000	0.6983
Total	0.3295	0.4892	0.4767	0.1623	0.6983	1.0000

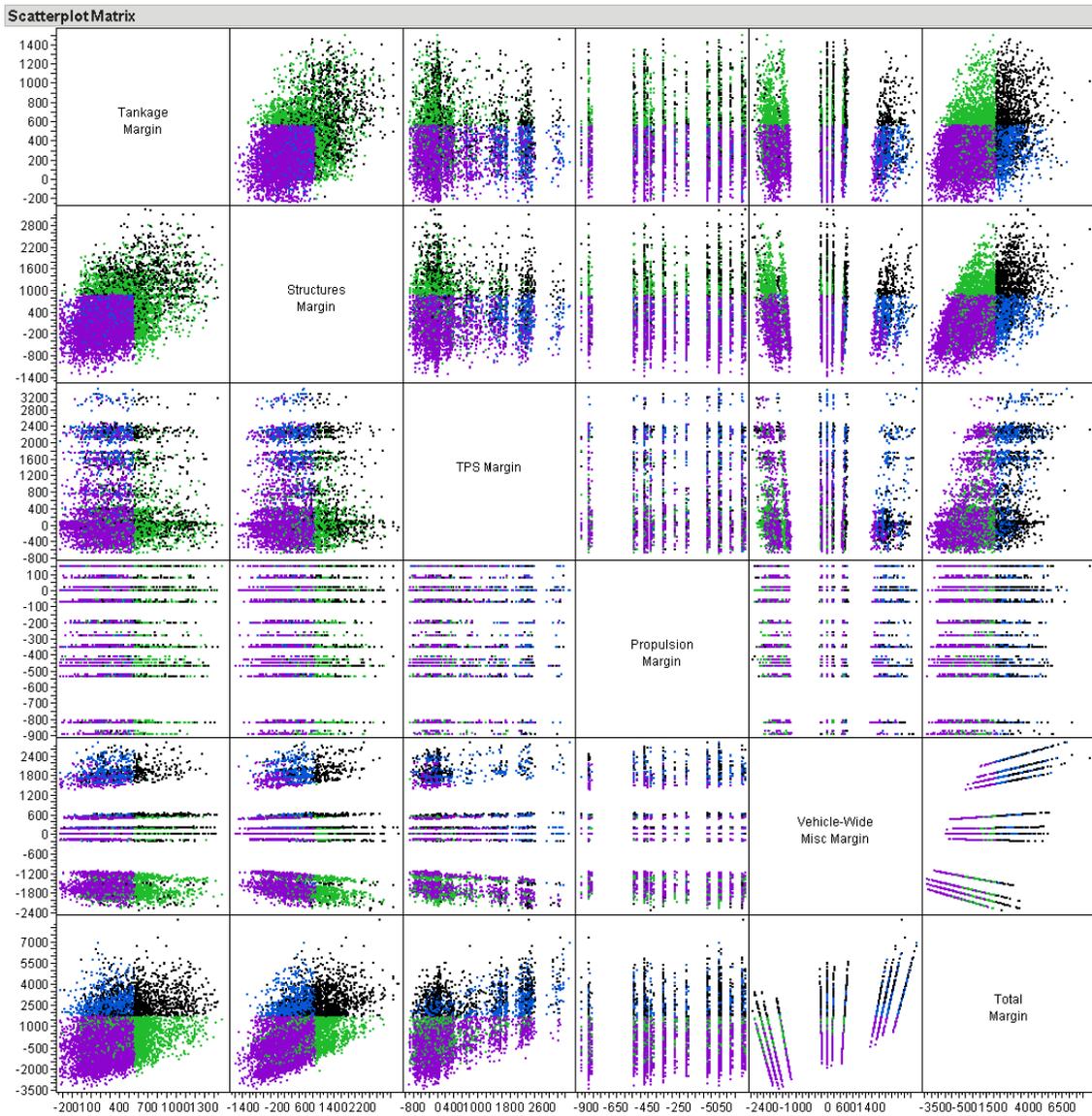


Figure 105: Scatterplot Matrix for Subsystem Margins Highlighting Thresholds

Table 21: Total Margin Estimation

	Expected Value (lbs)	80% Quantile (lbs)
Dry Weight	17,080	17,080
Range Estimating Margin	1,448	1,596
EMA Margin	481	1,711
Margin Sum	1,929	3,307
Predicted Dry Weight	19,009	20,387

8.4 Margin Selection

The final step of the hybrid methodology is to assign the final margin to be used by the program. As stated in Section 6.3, this can be done in two general methods: allocating a separate margin based on the results of range estimating and EMA separately or combining the simulation results into a single CDF. For this section, the analysis will take place on the entire data set with no excluded cases due to what-if scenarios; if a program decides to proceed with a what-if scenario, the analysis would proceed as hereafter described with the only change being the adjusted data set with excluded cases.

The first method of assigning a margin involves utilizing the results of range estimating and EMA separately. The results of range estimating are discussed in Section 8.2.4; these simulations resulted in an expected margin and 80% quantile margin of 1,448 lbs and 1,596 lbs respectively. Similarly EMA simulations resulted in an expected margin and 80% quantile margin of 481 lbs and 1,711 lbs respectively. As indicated in Table 21, the total predicted weight will have an expected value 19,009 lbs and an 80% quantile of 20,387 lbs. This represents a 11.3% margin for the expected value case and a 19.3% margin for the 80% quantile case.

Another way of allocating margins separately is by looking at individual subsystems. For this analysis, the outputs of range estimating are divided into the subsystem categories defined in Section 8.3.1; as the full WBS contains electrical subsystems not covered by these parameters, all other subsystems are listed in the “Other” category. The full breakdown of subsystem margin by both range estimating and EMA can be seen in Table 22. While the expected total margin is equivalent to the estimated margin obtained by looking at the

Table 22: Total Margin Estimation via Subsystems

Subsystem	Expected Value (lbs)	80% Quantile (lbs)
Dry Weight	17,080	17,080
Propellant Tanks (Range Estimating)	143	186
Propellant Tanks (EMA)	368	561
Structures (Range Estimating)	504	619
Structures (EMA)	383	876
TPS (Range Estimating)	123	157
TPS (EMA)	244	318
Propulsion (Range Estimating)	241	301
Propulsion (EMA)	-212	2
Other (Range Estimating)	435	500
Other (EMA)	N/A	N/A
Vehicle-Wide (Range Estimating)	N/A	N/A
Vehicle-Wide (EMA)	-303	471
Margin Sum (Range Estimating)	1,446	1,763
Margin Sum (EMA)	481	2,230
Predicted Weight	19,007	21,073

total weight, the 80% quantile is far more conservative.

Margin can also be analyzed by combining the Monte Carlo simulations. In order to do this, the range estimating and EMA simulation outputs were converted to deltas from the baselines weights. In the case of subsystem evaluations, they were converted to deltas from the underlying subsystem baseline weights. These deltas can be added together to result in a case-by-case delta from the baseline. These cases are then analyzed to produce a new CDF for weight and margin (Figures 106 and 107).

The results for margin analysis through a combined analysis can be seen in Table 23. The expected value for the predicted dry weight of the FAST RFS F is equivalent to all other estimations made by calculating the expected value. However, the 80% quantile value is slightly less than the previous estimate made by analyzing the total vehicle.

Similarly, the Monte Carlo simulation data for individual subsystems can be combined to produce total distributions for each subsystem. Table 24 shows the values for each combined subsystem. As expected, the expected value prediction is equivalent to the previous three methods of analyzing this data. However, the 80% quantile measurement is slightly different from the previous methods proving a less conservative estimate of margin when compared

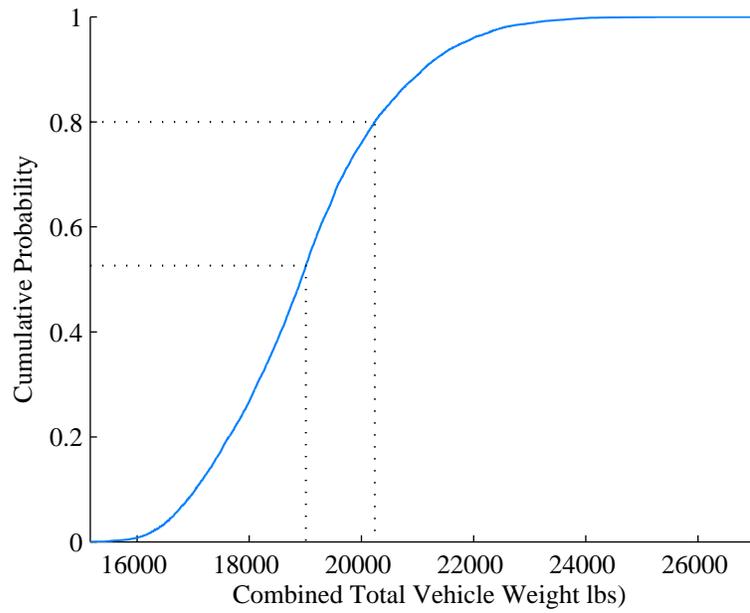


Figure 106: Monte Carlo Results for Combined Total Weight

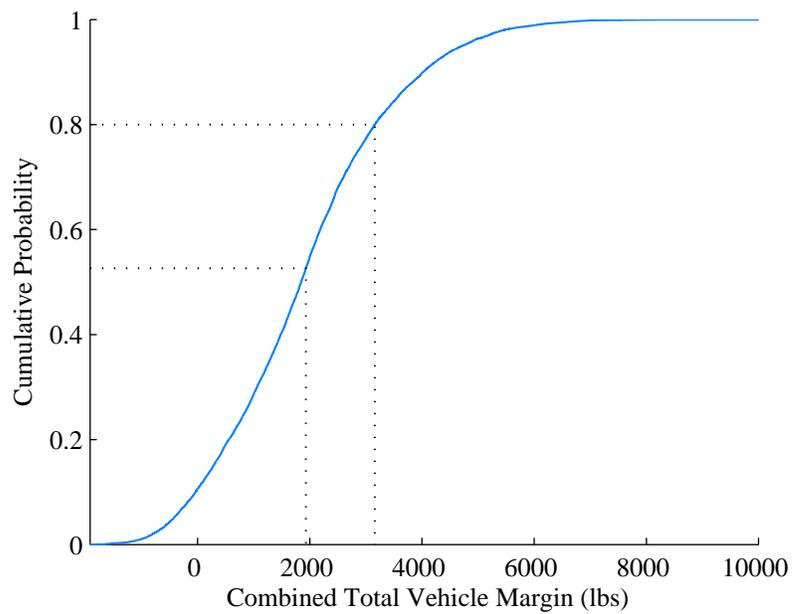


Figure 107: Monte Carlo Results for Combined Total Margin

Table 23: Total Margin Estimation

	Expected Value (lbs)	80% Quantile (lbs)
Dry Weight	17,080	17,080
Design Margin	1,929	3,157
Predicted Dry Weight	19,009	20,237

Table 24: Total Margin Estimation via Subsystems

Subsystem	Expected Value (lbs)	80% Quantile (lbs)
Dry Weight	17,080	17,080
Propellant Tanks	511	707
Structures	888	1,401
TPS	367	449
Propulsion	29	288
Other	435	500
Vehicle-Wide	-303	471
Margin Sum	1,929	3,819
Predicted Weight	19,009	20,899

to the previous subsystem-centric analysis method.

Ultimately, the main difference between these analysis techniques is not the final margin values but rather the ability for a decision maker to evaluate separate portions of the model output. Looking at EMA and range estimating results separately gives a different view of the problem to a decision maker compared to a simple CDF of total predicted weight. Similarly, breaking the problem down into subsystem estimates provides a different view than looking at the total predicted weight. Ultimately, it is up to the decision makers in the program under analysis to decide on which data views are most useful. This example problem is intended to show how these views can be used to make decisions.

8.5 Additional Use Cases for EMA

In addition to using EMA as part of the hybrid methodology to estimate a vehicle or subsystem design margin, a vehicle program office or subsystem office can use the EMA model as a way to validate the EMA model, examine sensitivities, explore design decisions, and perform risk mitigation. This section uses the FAST RFS F EMA model to explore

alternate use cases of EMA.

8.5.1 EMA Model Validation

After the creation of a vehicle-level EMA model, the program office will want to validate the output of the model versus the experience of the experts, design teams, and subsystem offices which contributed to the creation of the model. This use case will show how the structures subsystem office can validate the output of the structures portion of the EMA model.

This analysis starts by running a 10,000 case Monte Carlo simulation of just the structures portion of the FAST RFS F EMA model. The structures portion consists the structures subsystem matrix as defined in Figure 81; these conditions correspond to the structures portion of the WBS defined in Table 16 with the exception of three items: oxidizer tank insulation, landing gear, and wing fairing. To account for this difference, the output of the Monte Carlo simulation will have 1,913 lbs (the total baseline weight of the oxidizer tank insulation, landing gear, and wing fairing) added to the simulation results.

The first thing that should be looked at is the raw output of the Monte Carlo simulation. The structural subsystem weight CDF can be seen in Figure 108; the difference between the simulation output weight and the structures subsystem weight can be seen in Figure 109. The first thing that should be checked on these curves is that the baseline weight, 8891 lbs, is within the bounds of the CDF. As can be seen in weight difference curve (Figure 108), the value of 0 lbs appears as a valid value therefore the model encompasses the original baseline value. The next thing that can be seen in the curve is that the majority of cases represents mass growth, and relatively few cases represent mass savings. This result is also consistent with the construction of the model where most selections will result in mass growth while very few selections would result in a mass savings.

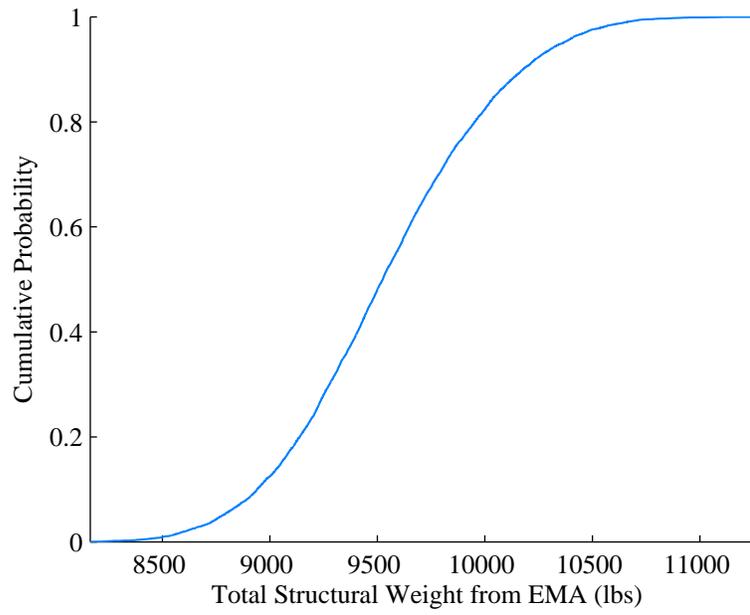


Figure 108: Monte Carlo Results for Structural Subsystems

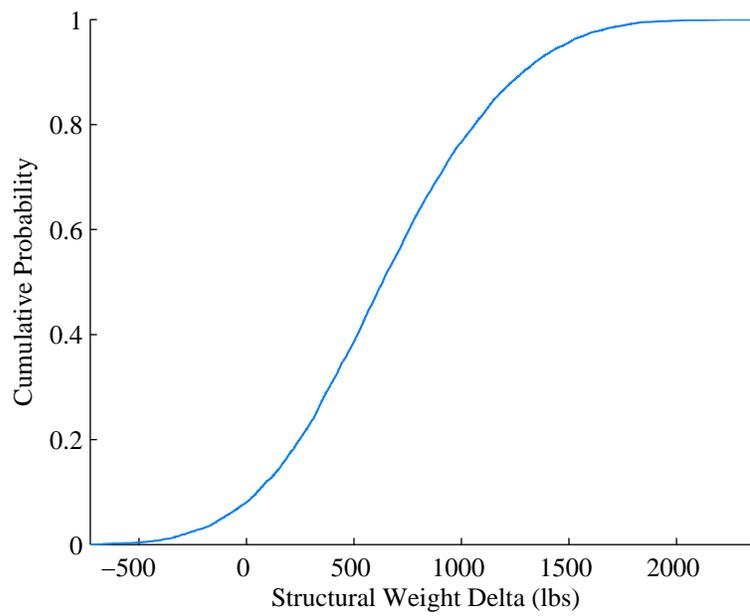


Figure 109: Monte Carlo Results for Structural Subsystem Margin

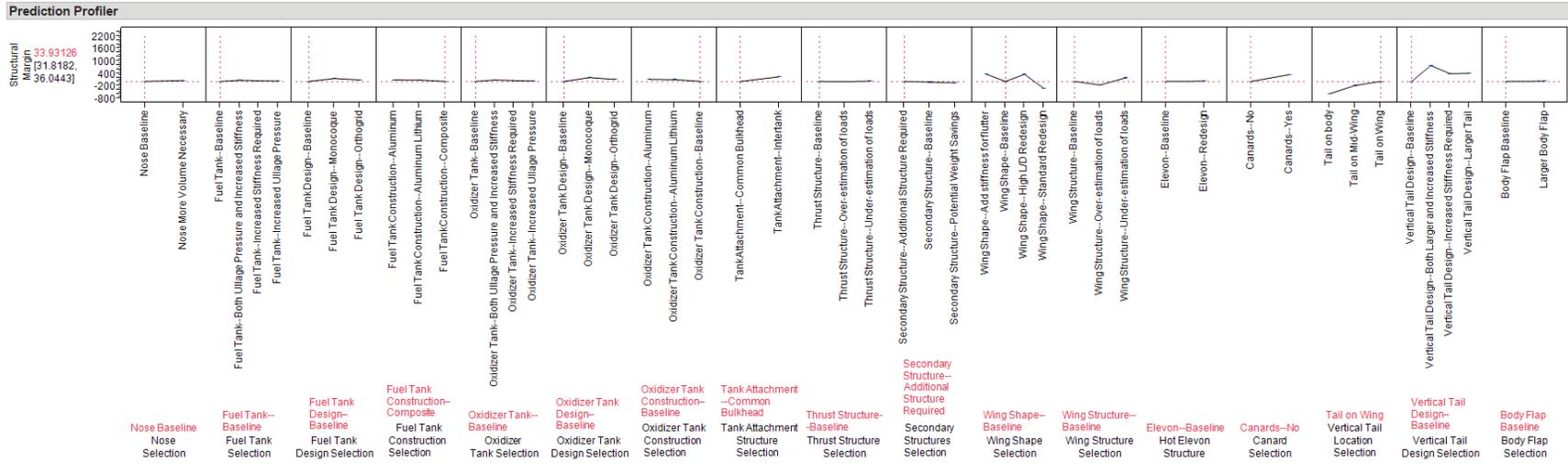


Figure 110: Prediction Profiler for Structural Subsystems-Baseline Requirements

Given this Monte Carlo simulation, the results can be regressed into a linear statistical model. [90] This model can yield a prediction profiler which visualizes the total derivative of each regression; this prediction profiler can be seen in Figure 110. [117] The slope of the lines in each box of the prediction profiler characterizes both the sensitivity and directionality of the weight response to the alternatives in the EMA model. This can provide insights into engineers looking to validate the EMA model because the derivatives are readily visualized; this enables an easy method of checking the model versus the intuition of subsystem engineers. For example, if the prediction profiler showed that a response thought to provide a weight savings actually incurs a weight penalty, then an engineer can flag that part of the model for further study.

Another view of the model which can be used for validation is a scatterplot matrix of the selected conditions; a scatterplot matrix plots every selected dimension versus every other selected dimension. A scatterplot matrix of the selected fuel and oxidizer tank conditions can be seen in Figure 111; this figure visualizes the frequency of each combination of selections—the density of each plot represents the relative frequency of selection. Visualizing this information allows for the validation of the original probability assignments and probability relationships. For example, the incompatibility between composite construction and orthogrid design can be validated using this view.

8.5.2 Subsystem Sensitivity

Another fundamental question which can be answered on a subsystem level is 'how sensitive is the baseline weight of the structures subsystem to baseline changes.' First, the global sensitivity can be determined by plotting the scaled estimates calculated during the linear regression. This plot can be seen in Figure 112. Based on these scaled estimates, the wing design choices and vertical tail have placement have the largest impact. The wing is the largest structural subsystem, therefore wing design choices will have large impacts. Similarly, the vertical tail can be placed on the wing tip or mid-wing; this placement will result in an overall higher wing structural weight to support the vertical tail.

The next level of sensitivity of use to a vehicle designer is the local sensitivity around

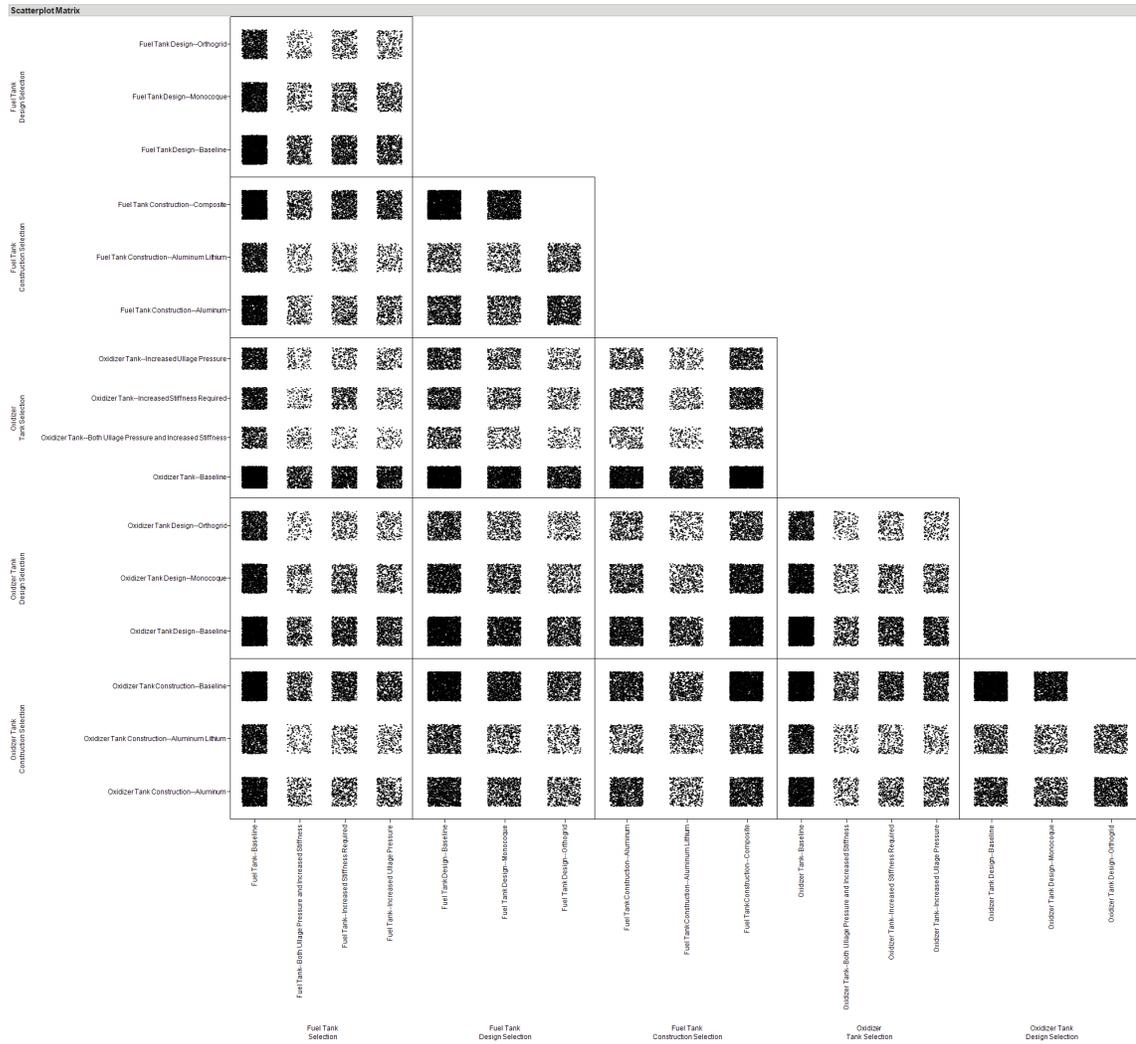


Figure 111: Scatterplot Matrix for Selected Fuel and Oxidizer Tank Conditions

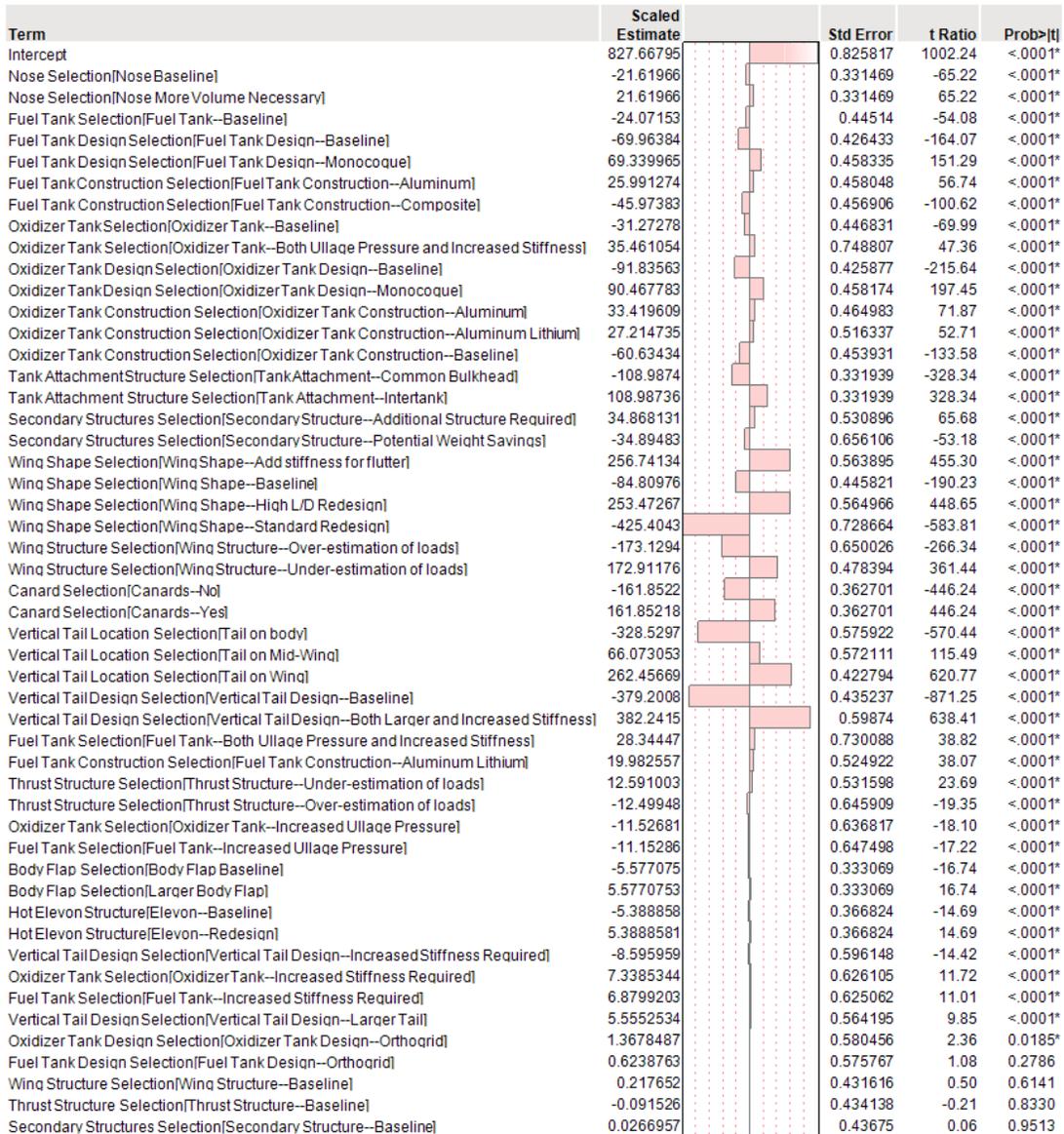


Figure 112: Tornado Plot for Structural Subsystem Design Choices

the baseline. This will show which excursions from the baseline will immediately impact the structural weight. These local sensitivities can be visualized using the prediction profiler in Figure 110. This prediction profiler illustrates local sensitivities because the baseline design is chosen for each potential input. This profiler shows that the current baseline design is very sensitive to changes in the wing shape, the addition of canards, vertical tail placement, and tail design changes. Propellant tank choices have lesser impacts. Finally, this prediction profiler shows that weight is insensitive to the secondary structures options.

8.5.3 Subsystem Design Decisions

The maturation of a vehicle from conceptual design to preliminary design is an iterative process in which higher-fidelity analyses motivate design decisions. The system design matures as this series of decisions is made. As shown in Section 2.3.1.3, this series of design decisions represents volitional uncertainty as risk managers must account for future design decisions. However, EMA allows for risk managers to examine the sensitivity of design decisions to the overall vehicle weight.

In order to examine design decisions, trends in acceptable designs must be observed. Figure 113 shows a scatterplot matrix of propellant tank options, and Figure 114 shows a scatterplot matrix of wing and vertical tail options. In each figure, blue dots represent designs which would not meet the 80% quantile structures and propellant tank margin of 1,437 lbs as defined in Section 8.3.5.1. These blue dots are very hard to see in Figure 113 because they are almost uniformly scattered throughout each box; this uniformity shows that propellant tank design decisions are insensitive to meeting the set margin. However, Figure 114 shows a high clustering of blue dots defining vertical tail placement; this indicates that the placement of the vertical tail on the wing is a high-risk design decision. This risk is further compounded if the original analysis in sizing the tail and wing structures

Knowing this information, a vehicle office may want to hold off on making high-risk design decisions until further study can be completed. For this example, a vehicle office can decide to go forward with composite designs for both the fuel and oxidizer tank as well as the baseline wing. An updated CDF with these designs chosen can be seen in Figure 115.

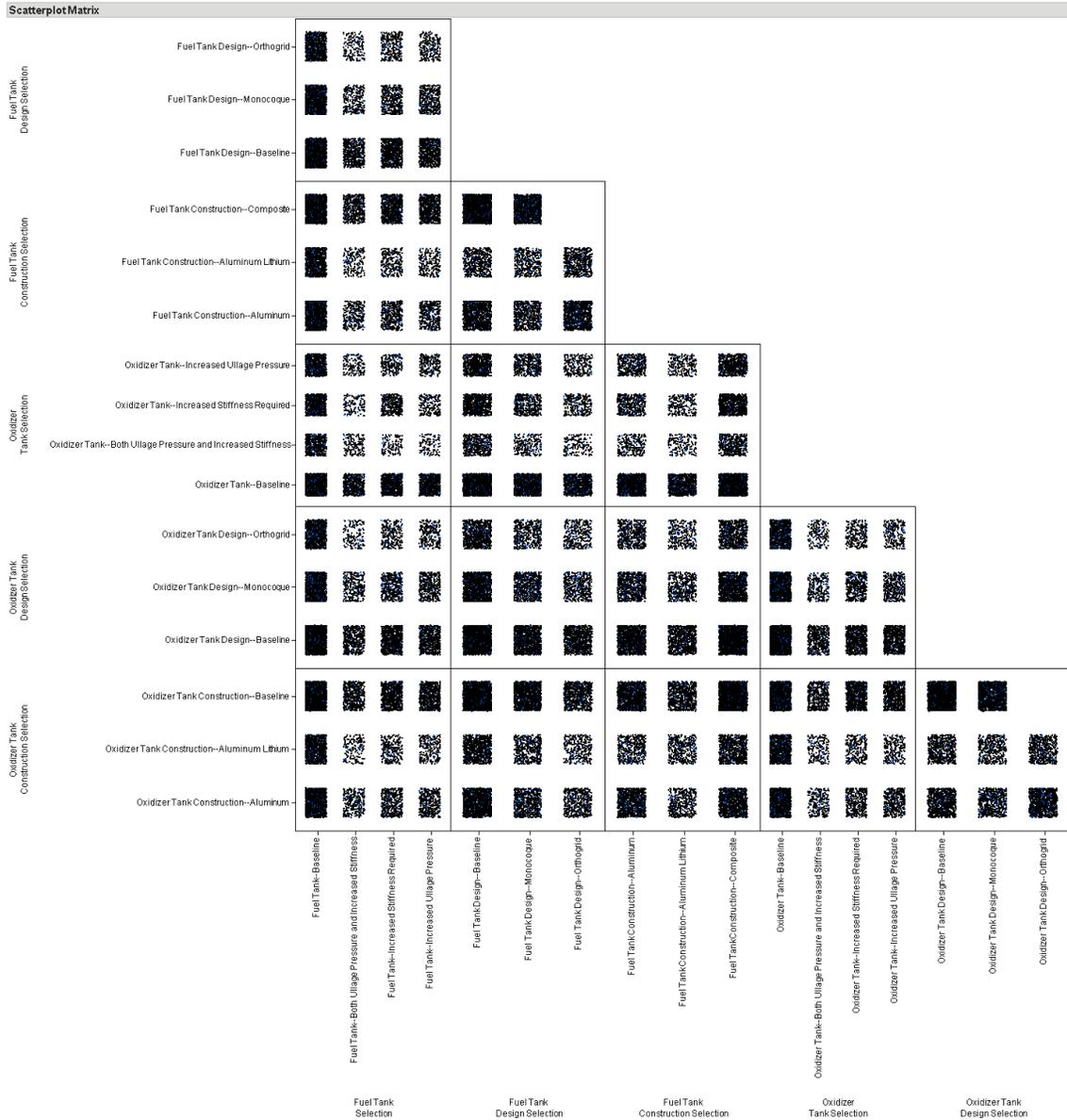


Figure 113: Scatterplot Matrix for Structures Subsystem Condition Selections Highlighting Cases With Mass Growth of More than 1,437 lbs

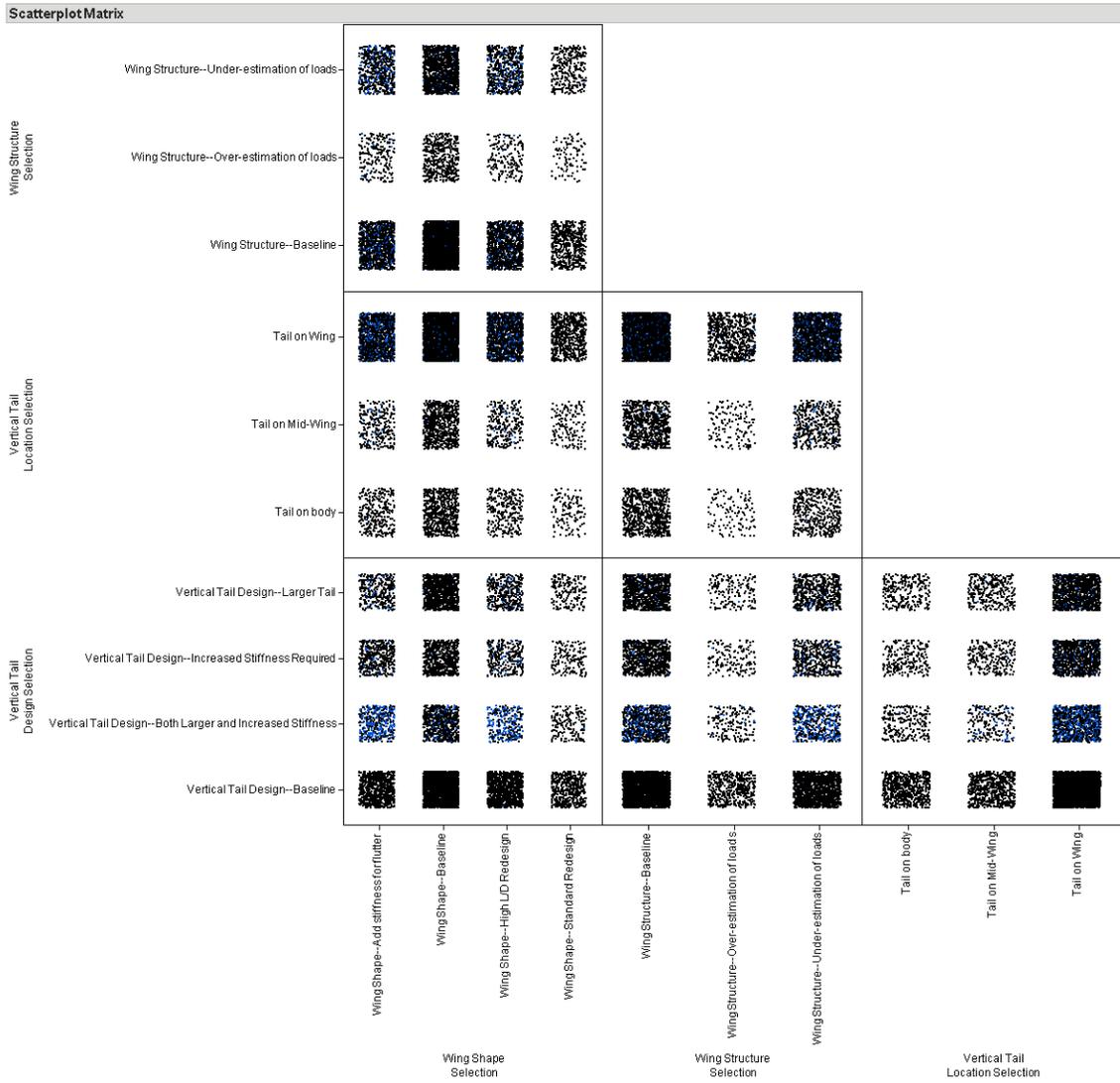


Figure 114: Scatterplot Matrix for Structures Subsystem Condition Selections Highlighting Cases With Mass Growth of More than 1,437 lbs

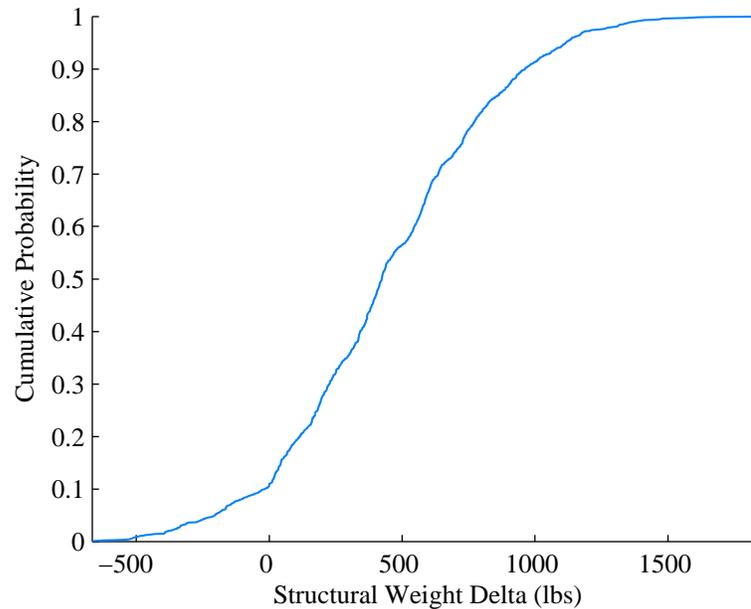


Figure 115: Structural Subsystem Margin Having Decided on Composite Tanks and Baseline Wing

These decisions marginally reduce uncertainty while maintaining the option to place the vertical tail on the body or mid-wing in order to save weight.

Alternatively, a vehicle office could decide to buy extra design margin by moving the vertical tail to the body. The CDF for these cases can be seen in Figure 116. This CDF shows a much greater probability of producing a vehicle within the original specified design margin. Furthermore, a program could then decide to proceed with lower-cost options for other portions of the vehicle such as the propellant tanks, thrust structure, and secondary structures.

By performing trades such as those shown in this section, EMA can be used as a way to map out design decisions and determine which system analyses are on the critical path. By addressing volitional uncertainty, this will ultimately give more information about the vehicle's sensitivity to design decisions to risk managers and can lead to more successful development programs.

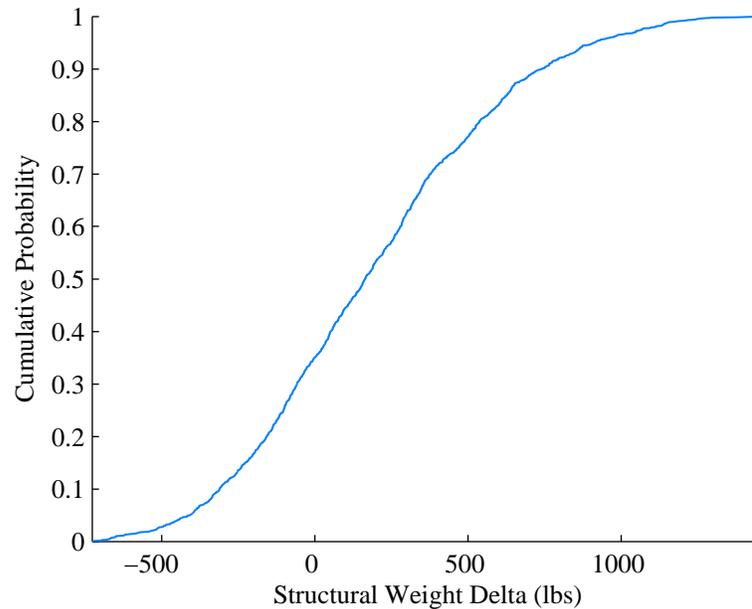


Figure 116: Structural Subsystem Margin Having Decided on Vertical Tail Placement on Body

8.5.4 Subsystem Risk Analysis

8.5.4.1 6 G-limit Requirements Change

The structural subsystem can also be subject to requirements uncertainty as defined in Section 2.3.2.1. In this model, the primary requirement which defines potential structural weights is the g-limit requirement. EMA can be used to examine a scenario in which the 6 gee limit is chosen as the requirement.

A separate 10,000 case Monte Carlo simulation was conducted for structural subsystem conditions in which the 6 gee limit was selected. The CDF for the total structural weight and weight difference from baseline can be in Figures 117 and 118 respectively. The first thing that can be observed is that the baseline weight of the subsystem is not present on either curve. Furthermore, the program has less than a 20% chance of producing a vehicle which weighs less than or equal to the assigned structural subsystem margin of 1,437 lbs as defined in Section 8.3.5.1.

In order to look at risk mitigation options, a vehicle manager can select all of the design

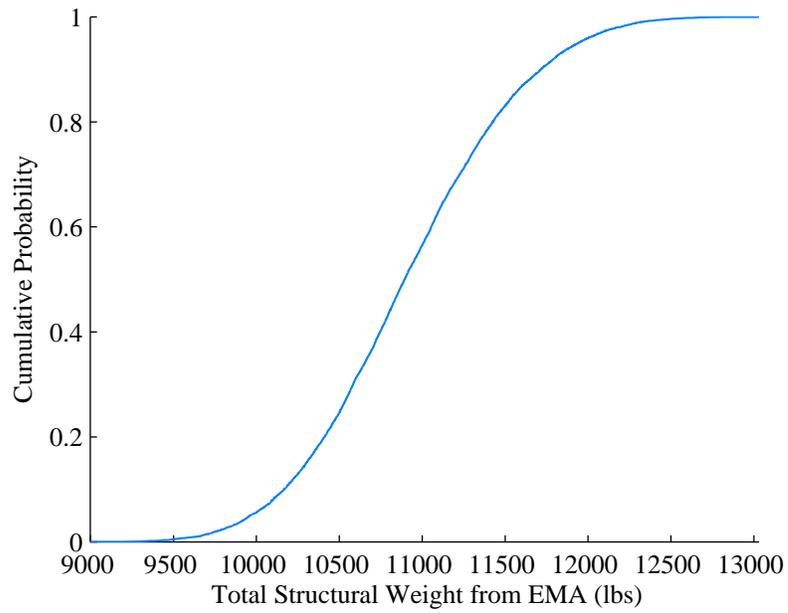


Figure 117: Monte Carlo Results for Structural Subsystems, 6 g Limit

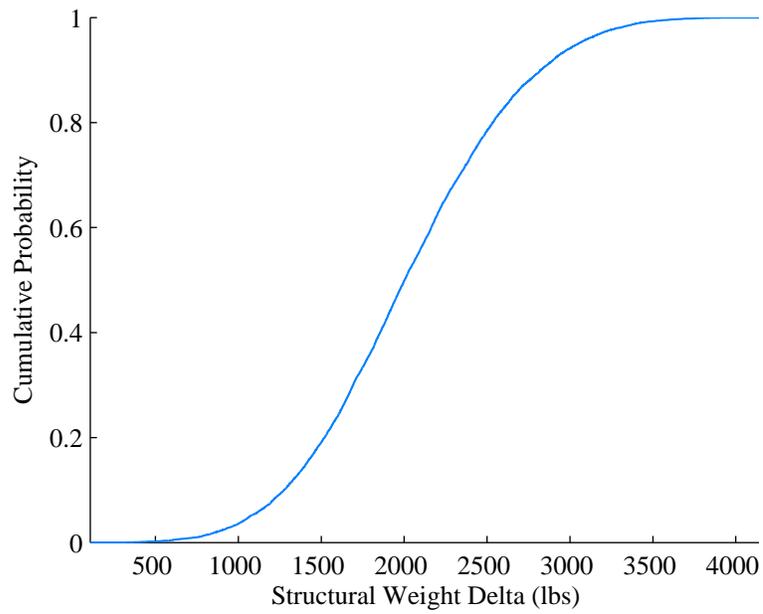


Figure 118: Monte Carlo Results for Structural Subsystem Margin, 6 g Limit

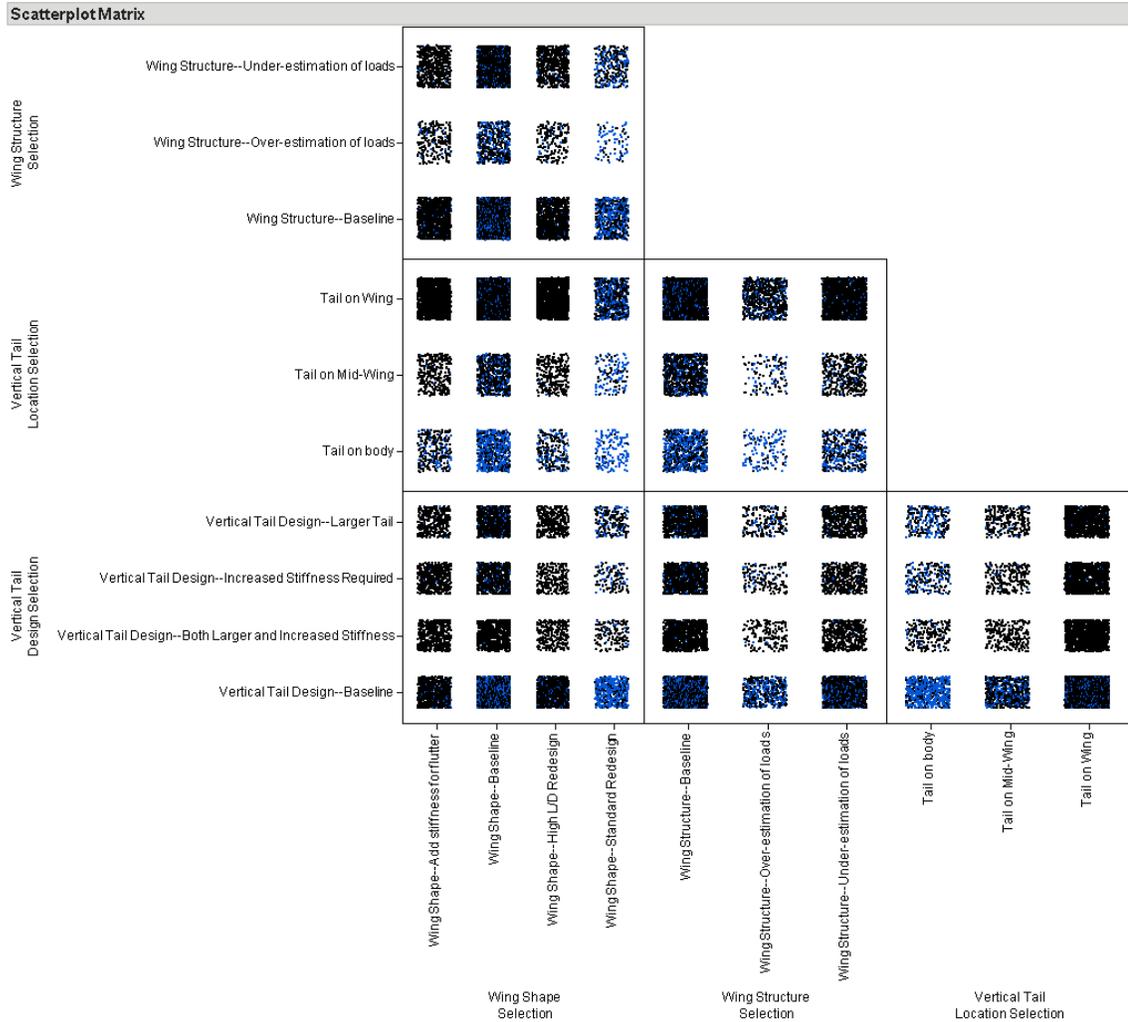


Figure 119: Scatterplot Matrix for Structures Subsystem Condition Selections Highlighting Acceptable Designs, 6 g Limit

selections which meet the original design margin; cases which meet the 1,437 lb design margin are highlighted in Figure 119. This scatterplot shows the wing and tail design options. This figure shows that very few baseline wing and tail configurations will meet the margin. In order to mitigate risk, a change to the baseline must be made. Two options are apparent. As in Section 8.5.3, a mass savings can be found by moving the vertical tail to the body or mid wing. Another mitigating choice which can preserve a wingtip mounted wing is changing the wing shape from its current high L/D configuration to an earlier 'standard' wing considered in earlier iterations of the RFS F design.

8.5.4.2 Composite Technology Failure

Because the structural subsystem is the focus of technology development programs, it is subject to technology uncertainty as defined in Section 2.3.1.3. Furthermore, it is subject to discrete technology uncertainties which are unable to be modeled through range estimating. This discrete technological uncertainty takes the form of the failure of the composite propellant tank and composite common bulkhead programs. This scenario can be modeled through EMA.

A 10,000 case Monte Carlo simulation is run through structural conditions of the FAST RFS F EMA model. Before each run, the 'non-inclusion' conditions for both the composite tank and common bulkhead technology development programs were selected triggering an incompatibility with the corresponding vehicle design options. The CDF for the total structural weight and weight difference from baseline can be in Figures 120 and 121 respectively. Figure 122 shows the difference between the baseline model run and the run eliminating composite technologies; the elimination of composite options shifts the entire curve slightly to the right of the baseline and shows the sensitivity of the structural subsystem to this technology uncertainty.

8.5.5 Requirements Risk Mitigation

A similar sensitivity analysis to the one carried out in Section 8.5.2 can be carried out for the entire vehicle. This can identify which condition selections have the overall highest impact on the RFS F vehicle weight. The tornado plot for the total sensitivity analysis can be seen in Figure 123; this plot clearly shows that the transverse G-limit and single-stage vehicle performance have the highest overall impact on vehicle weight. This is expected as these top-level requirements will drive the required performance of subsystems.

Knowing that more stringent G-limit and single-stage vehicle performance requirements drive the overall system weight, decision makers may decide to allocate resources other than weight toward mitigating these risks. In order to determine if allocating program resources is worth a mass savings, what-if scenarios need to be played out in order to determine the impact on the vehicle weight.

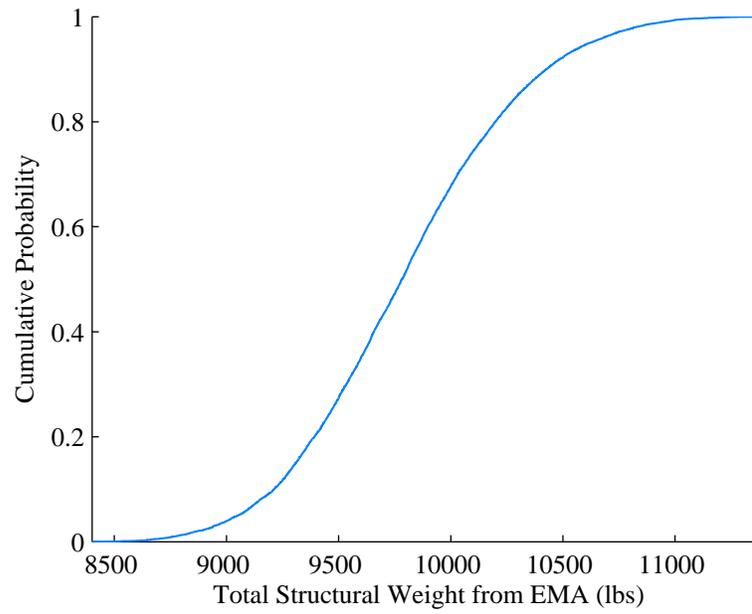


Figure 120: Monte Carlo Results for Structural Subsystems, Composite Technologies Eliminated

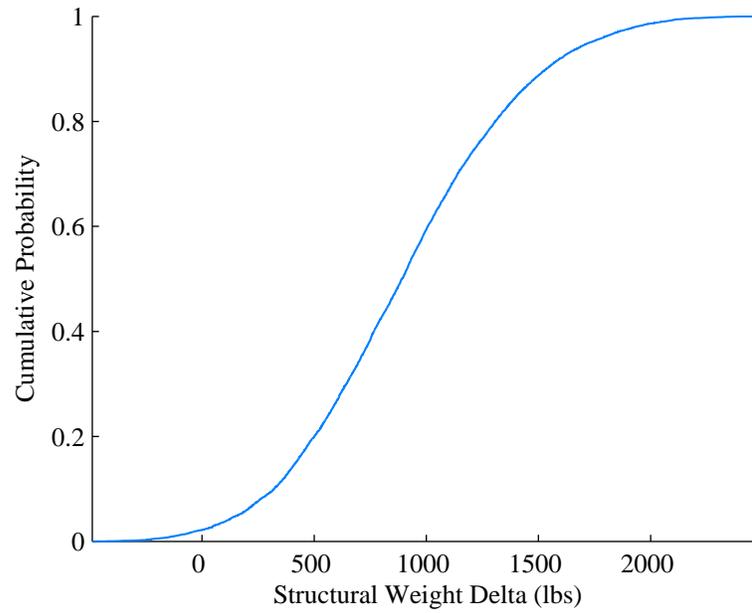


Figure 121: Monte Carlo Results for Structural Subsystem Margin, Composite Technologies Eliminated

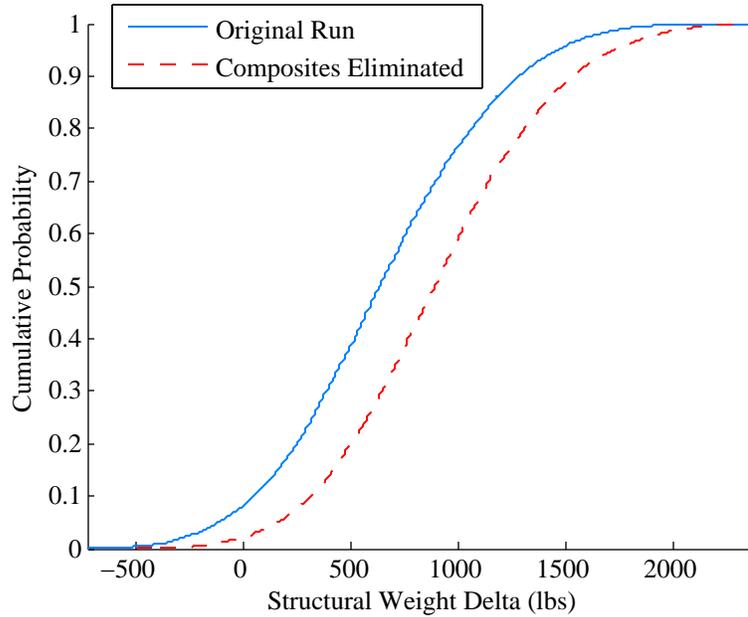


Figure 122: Comparison Between Baseline Monte Carlo Simulation and Eliminated Composite Technologies

This scenario is analyzed by performing through filtering the Monte Carlo output data. All output cases which feature either a 6 g requirement or an increased single-stage vehicle performance threshold are highlighted in red in Figure 124. As can be seen in this figure, these highlighted cases contain most of the highest required margins as well as most of the highest tankage and structures subsystem margins. Overall, there are 1,874 cases out of the original 10,000 case Monte Carlo which have these heightened requirements. The remaining 8,126 cases provide an adequate sample size for probabilistic analysis.

The probabilistic results for total weight and margin for the filtered Monte Carlo simulation can be seen in Figures 125 and 126 respectively. A scenario in which program management allocates resources and avoids increased G-limit and single stage performance requirements results in an expected weight of 17,185 lbs and an 80% quantile weight of 18,201 lbs. This results in an expected margin of 105 and an 80% quantile margin of 1,121 lbs. This scenario has the effect of trimming 380 lbs from the expected margin and 475 lbs from the 80% quantile margin as defined in Table 21.

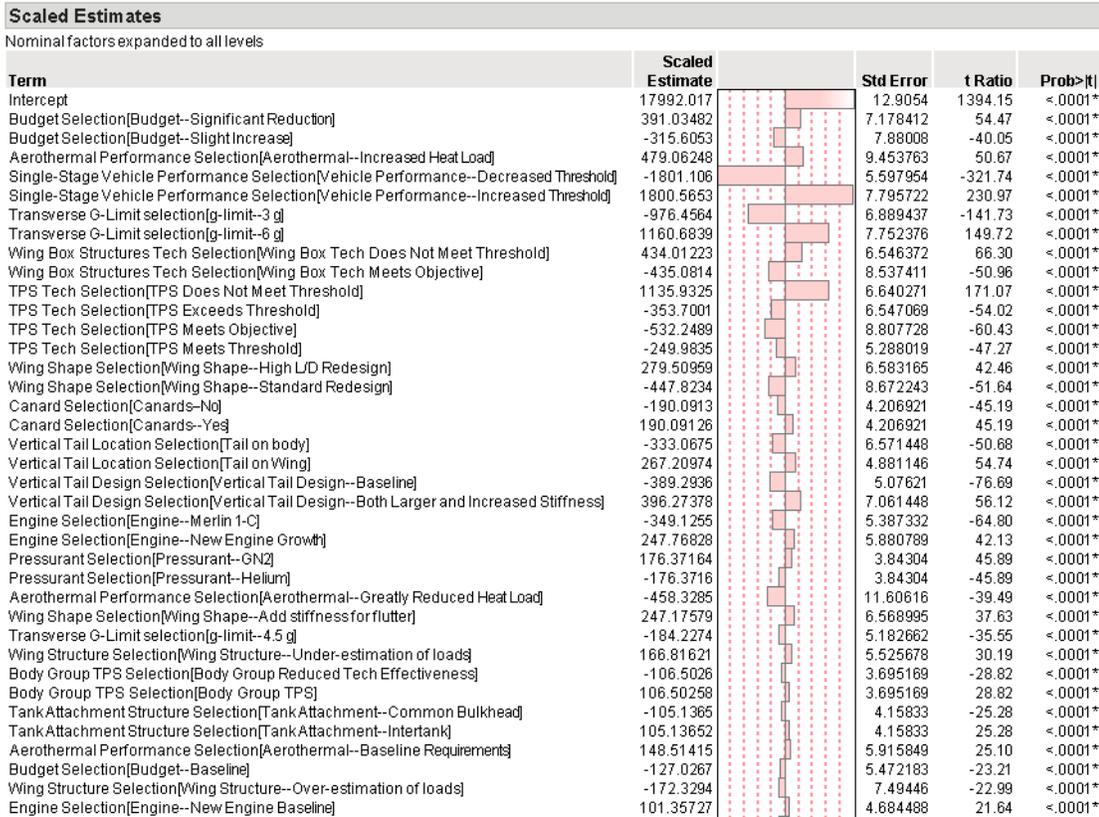


Figure 123: Prediction Profiler

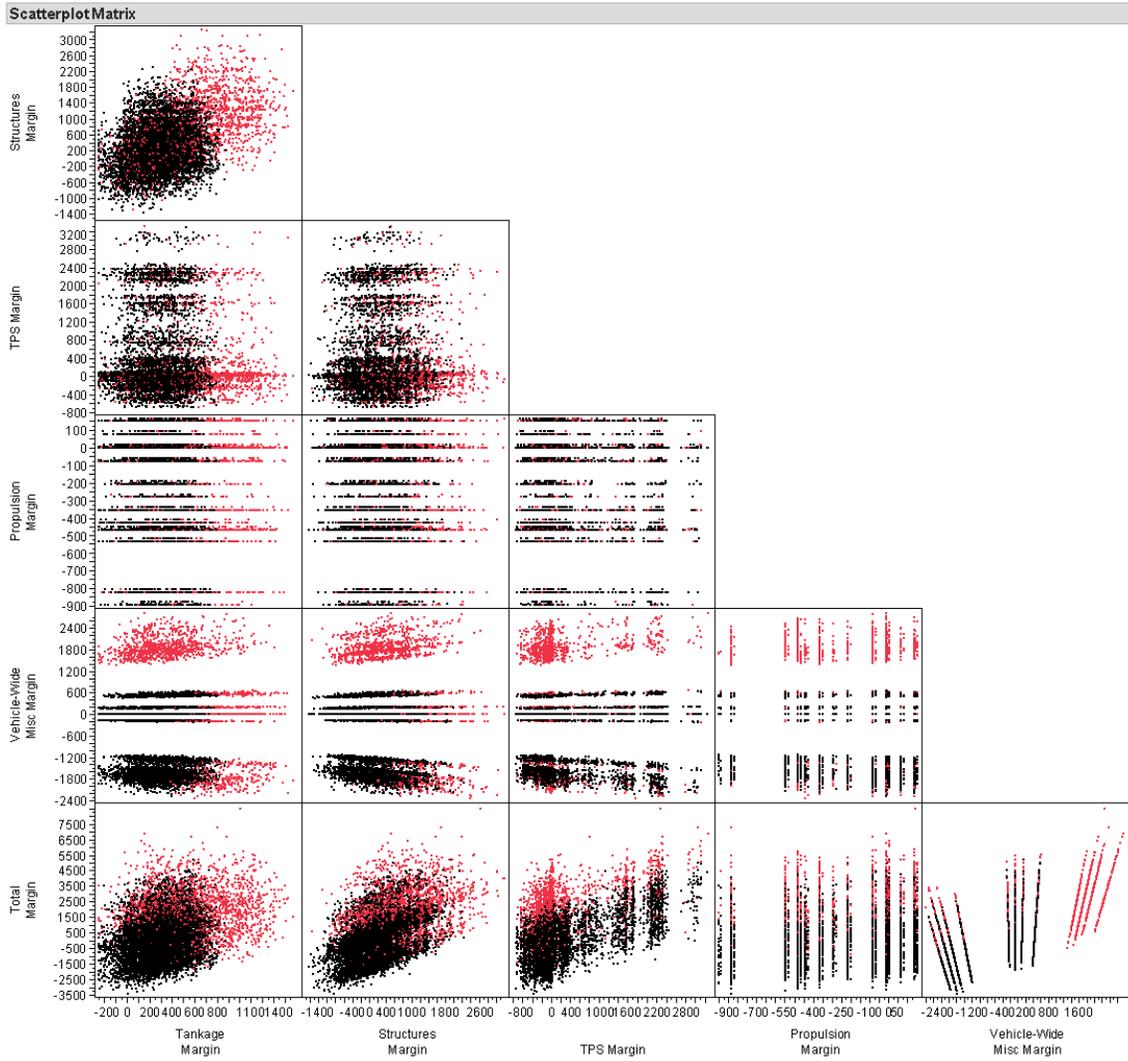


Figure 124: Scatterplot Matrix with 6g Limit and Increased Single-Stage Vehicle Performances Cases Highlighted

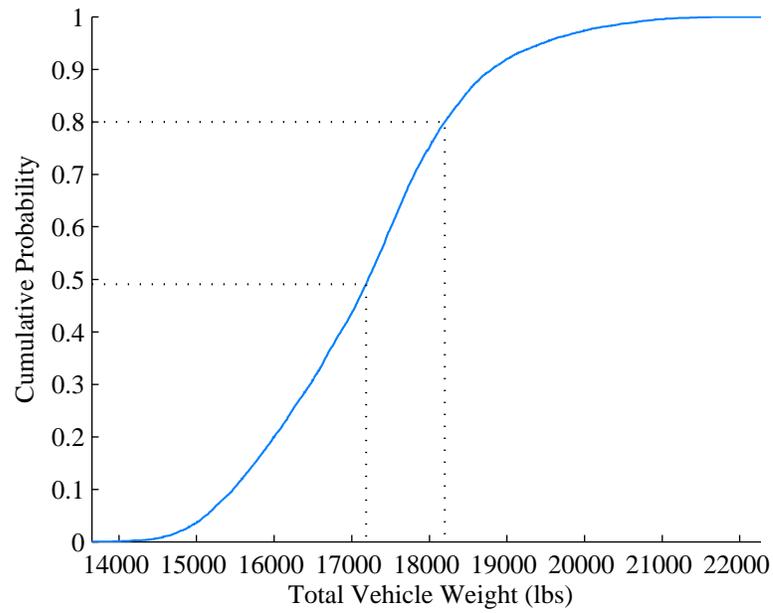


Figure 125: Monte Carlo Results for Total Weight Excluding Demanding Requirements

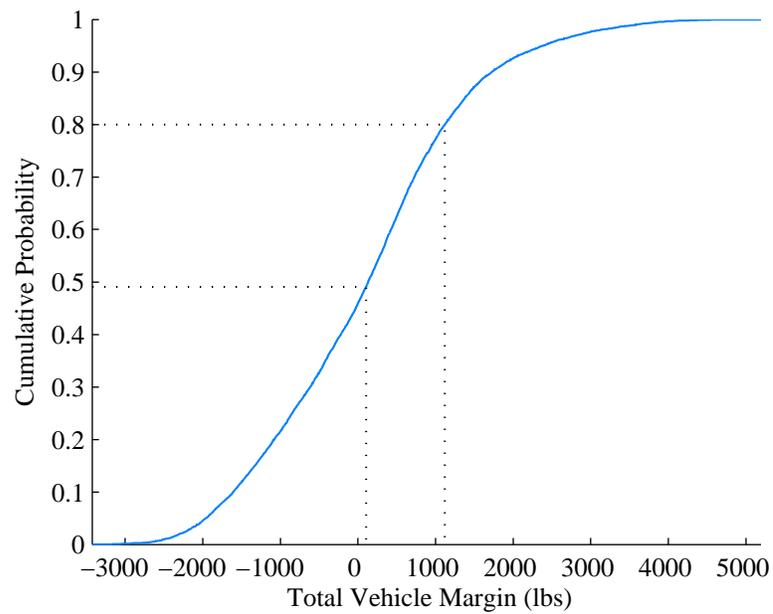


Figure 126: Monte Carlo Results for Total Margin Excluding Demanding Requirements

This section illustrated an example of a single what-if analysis. This analysis should be repeated for many what-if scenarios to determine the best use of a program's risk-mitigating resources. While this analysis can provide insight into potential weight savings as a result of risk-mitigating efforts, ultimately it is up to decision makers to decide on appropriate mitigating strategies.

CHAPTER IX

CONCLUSIONS AND FUTURE WORK

9.1 Conclusions

This work began by examining the impact of mass growth on space and launch vehicles; specifically, its effects on novel concepts. This examination shows that the physics of space launch leads to an exponential decline in technical performance due to mass growth, and many x-vehicles and novel concepts were either canceled, delayed, or otherwise hampered by mass growth. Furthermore, in order to mitigate mass growth, the uncertainty surrounding the final flight mass must be predicted at a very early stage in the design process. These facts motivated the research objective of this thesis.

Research Objective: to create a methodology for estimating the overall system technical performance uncertainty through the forecasting of potential performance shortfalls for a novel concept such as a reusable booster system

In order to begin addressing this research objective, a literature review on uncertainties was conducted. This led to an examination of the uncertainties which affect space and launch vehicles and a subsequent taxonomy of uncertainties affecting development programs. This understanding of uncertainty would play a central role in the evaluation of methods used to evaluate weight uncertainty and ascertain design margin in space and launch vehicles.

Of the four methods that are commonly used to address uncertainty and estimate a design margin, two methods, predetermined percentage and historical regression, are ill-suited to the problem of evaluating a novel concept because of data requirements. Of the remaining two methods, expert opinion and range estimating, range estimating was selected as a prospective method for extension because its shortcomings stem from not addressing all forms of uncertainty. Specifically, a bottom-up estimation technique like range estimating cannot account for baseline changes. These observations led to the central research hypothesis which outlines the hybrid forecasting methodology developed in this

thesis.

Hypothesis 1: If a hybrid process is adopted with a bottom-up and a top-down component, then the relevant sources of uncertainty affecting space and launch vehicles will be quantified to enable a more complete estimate than can be constructed via existing methods of design margin estimation.

This hybrid methodology merges two separate styles of analyzing a development program's uncertainties. Because range estimating has been established as a mature methodology in multiple fields, it will be used in the hybrid methodology as the bottom-up analysis. Because range estimating is being used unmodified, the new focus of research focuses on creating a top-down methodology which accounts for baseline changes without double-counting estimates from range estimating.

Two distinct analyses need to occur in order to evaluate the effect of prospective baseline changes to the weight of a vehicle. These changes need to be both identified and quantified. These are the two gaps which must be addressed in the development of a new top-down analysis methodology. Furthermore, there is a dependency in these two gaps: prospective changes must be identified before they can be quantified.

First, the baseline changes need to be identified. This led to an investigation of scenario analysis as a way to identify prospective scenarios which would in turn lead to discrete baseline changes. However, traditional scenario analysis as described in literature only leads to the creation of small numbers of scenarios. Further literature search led to the usage of morphological analysis for scenario generation. Morphological analysis is powerful because it enables the generation of large numbers of alternatives in a way that is both explorative and exhaustive. Based on this literature review, it is decided that morphological analysis should form the basis for the top-down assessment methodology; this is formalized in Hypothesis 2.

Hypothesis 2: If a morphological approach to scenario generation is taken then large numbers of possible alternatives for the future baseline configurations can be identified.

The selection of morphological analysis solves the first of the two analysis gaps: the identification of prospective baseline changes. Next, attention is focused on the second gap

of quantifying these alternatives. Because morphological analysis produces a large number of potential alternatives, it is infeasible to individually analyze each prospective alternative. Therefore, an extension to traditional morphological analysis must be developed to enable the rapid analysis of alternatives. This idea leads to Hypothesis 3.

Hypothesis 3: If a matrix of alternatives can be extended to include quantitative information about alternative baseline configurations and potential futures, then the sources of uncertainty not addressed by range estimating can be quantified.

This hypothesis is addressed by the creation of Executable Morphological Analysis (EMA) as an extension from traditional morphological analysis. The central idea behind EMA is that by including small pieces of information about each condition in the morphological field, then quantitative information can be easily extracted from each combination.

EMA is composed of two data structures: the executable matrix of alternatives and the relationship matrix. The executable matrix of alternatives is an extension of the morphological field. In this extension, each condition contains attributes; specifically it must contain attributes which describe the condition's effect on the model's output and the condition's likelihood of being selected. Similarly, the relationship matrix is an extension of the cross-consistency matrix. The relationship matrix contains relationships between the conditions of the model and implements incompatibilities.

For the specific implementation of EMA for addressing the problem of forecasting margins, each condition will have three pieces of information: a likelihood, a multiplicative effect, and an additive effect. The likelihood value is a specified chance of a condition being selected relative to the other conditions in the same parameter; these likelihood values are constrained so that the sum of all likelihood values in a single parameter is equal to unity. The multiplicative effect acts as a scalar multiplier to the total dry weight of a vehicle, and the additive effect acts as a simple addition to the dry weight.

This extension establishes the data structures of EMA. However, data structures alone are insufficient to address Hypothesis 3. EMA must be implemented as a computerized model for rapid analysis, and the analysis algorithms must be determined. These two, lower-level gaps in the implementation of EMA are formalized in Hypotheses 3a and 3b.

Hypothesis 3a: A Monte Carlo simulation run over an Executable Matrix of Alternatives will yield an equilibrium distribution which describes the impact of the alternative combinations derived from the morphological field. This equilibrium distribution will require less computational resources to compute compared to a total sum of all potential combinations.

Hypothesis 3b: The inclusion of data, constraints, and complex relationships should be implemented using an object-oriented approach.

Hypothesis 3a was tested through Experiment 1—conducting a series of Monte Carlo simulations while varying the size of the executable matrix of alternatives and the interconnectedness of the relationship matrix. This experiment found that the number of Monte Carlo simulation cases necessary to generate accurate probabilistic results is relatively constant with the size of the executable matrix of alternatives and the complexity of the relationship matrix. For the largest fields tested, the expected value and 80% quantile measurements converged in 1250-1500 Monte Carlo cases. This result confirms Hypothesis 3a and shows that useful quantitative information can be extracted from an executable morphological analysis without significant computational effort.

Hypothesis 3b was substantiated through the object-oriented design and implementation of the data structure. Because EMA requires state-dependent functionality and new attributes and methods in each level of the data structures, the object-oriented framework acts as an enabler of EMA and represents an advancement over matrix-based implementations of traditional morphological analysis. Furthermore, the object-oriented framework can easily enable future applications of EMA through subclassing the abstract classes.

Together, the substantiation of Hypotheses 3a and 3b show that the data structures of EMA can be implemented as a model to extract probabilistic information. This in turn partially substantiates Hypothesis 3 in that a matrix of alternatives can in fact be extended to provide quantitative information.

With EMA data structures and algorithms substantiated through experiment and implementation, the focus of the research shifts toward proving that EMA is a useful forecasting tool. Hypotheses 1, 2, and 3 will be addressed through sample problems. The first sample problem seeks to use original source material to produce a weight forecast of the Space

Shuttle Orbiter, and the second sample problem seeks to demonstrate all aspects of the hybrid analysis methodology on a novel concept.

The Space Shuttle Orbiter problem tests the use of EMA as a method of forecasting baseline changes. By using a predetermined percentage to model in-scope mass growth, this experiment isolates the predictive effect of EMA on forecasting mass growth associated with baseline changes. In order to construct the EMA model for this problem, the historical record of proposed Orbiter designs was used to create a morphological field. The mass-impacts of each potential baseline change was determined by utilizing original mass properties reporting documents from the Orbiter program; the use of estimates based on original documents mitigates hindsight bias and ensures that this experiment produces a new forecast based on the input of 1970's experts. The output of the EMA model produced very good predictions of the Orbiter's final flight weight. This experiment fully substantiates Hypothesis 3 in that the uncertainties associated with baseline changes are accounted for in this method. Furthermore, Hypotheses 1 and 2 are partially substantiated because a form of a hybrid methodology utilizing morphological analysis was able to produce useful forecasts.

In order to fully substantiate Hypotheses 1 and 2, demonstrate the completion of the research objective, and address the original motivation of this thesis, a sample problem utilizing the complete hybrid methodology is used to estimate the design margin for a novel concept— the FAST RFS F technology demonstration vehicle. This sample problem utilized mass properties data from FAST program contractors to inform a weight breakdown structure. This weight breakdown structure formed the basis of a range estimating analysis. This range estimating analysis was augmented with an extensive EMA model; the EMA model for this sample problem was broken into three separate sections: programmatics, technology development programs, and vehicle alternatives. The Monte Carlo simulation output of this EMA model was used to conduct a probabilistic analysis on the total vehicle and subsystem weights as well as a sensitivity analysis and a what-if analysis. Finally, the results of both range estimating and EMA were used to examine the weight growth of the FAST RFS F.

The FAST RFS F sample problem demonstrates the use of morphological analysis to generate alternative vehicle baseline scenarios and demonstrates the use of a hybrid method with both a top-down and a bottom-up analysis component for analyzing vehicle development programs. Therefore, both Hypotheses 1 and 2 are substantiated. Finally, this sample problem demonstrates the use of EMA and range estimating as a way to quantitatively analyze uncertainties surrounding a novel project; this demonstration shows that the original research objective has been satisfied and that a hybrid analysis method consisting of range estimating and EMA is a useful tool for analyzing uncertainties and assigning a design margin.

9.2 Future Work

The first area of future work is derived from the generalized development of EMA. EMA, as developed in this thesis, is primarily a forecasting technique looking at specific baseline changes. However, EMA was designed to be a much more extensible methodology; the OO design allows it to be easily extended into other problem areas. For example, a different extension of EMA could be used to evaluate high-level architecture design problems for which there is no viable modeling and simulation solution.

The next area of future work involves re-evaluating decisions made in the current formulation of EMA. Specifically, the formulation of two-way relationships and one-way relationships. The decision to have a hierarchy of matrices to limit the number of one-way relationships was made to minimize the time spent in workshops performing pairwise comparisons of conditions. However, EMA models would be far more flexible if all relationships were one-way and two-way relationships were a special type of relationship. This requires more study and implementation on sample problems in order to determine the best practices for usage in a risk/opportunity workshop.

The final area of future work is extending the existing form of EMA to account for schedule and cost. This extension would only require adding cost and schedule attributes to the conditions of an EMA model. This extension would make the methodology far more powerful; decision makers would be able to trade off mass properties, cost, and schedule

when evaluating margins for all of the above. This would also enable better risk mitigation strategies by more easily identifying appropriate alternative development paths.

9.3 Contributions

This work provided several contributions to the fields of uncertainty analysis and aerospace systems engineering. The first contribution is the creation of a taxonomy of uncertainty for vehicle development programs which distinguishes between exogenous and endogenous sources of uncertainty. This is a key feature over previous taxonomies of uncertainty which makes it more relevant to a development program office. By making a key distinction between exogenous and endogenous sources of uncertainty, this taxonomy focuses thought on the distinction between uncertainties inside and outside the influence of a program office. This distinction should enable a better understanding of uncertainties and allow for the employment of better mitigation strategies.

Another contribution of the taxonomy of uncertainty is its application to methods of forecasting margin. This taxonomy provides a direct explanation for the shortcomings of range estimating as a method to quantify uncertainties. Furthermore, the taxonomy provides a theoretical basis for the hybrid analysis methodology used to evaluate vehicle development programs.

The next contribution is the extension of traditional morphological analysis to create EMA. The idea of representing each condition of a morphological field as an object greatly extends the capability of morphological analysis. This extended capability not only allows the enumeration of compatible alternatives but also allows for the quantitative analysis of alternatives and the total alternatives space. The object-oriented implementation developed in this thesis also enables more robust morphological models through the creation of state-dependent relationships and dynamic constraints.

After the extension of morphological analysis into EMA, EMA was then applied to the specific forecasting problem of predicting a space or launch vehicle's flight weight. EMA allows the evaluation of a large number of alternative future scenarios; additionally, experimental evidence shows that most sizes of an executable matrix of alternatives can be

evaluated through an inexpensive Monte Carlo simulation. Furthermore, this forecasting methodology was shown to effectively account for baseline changes in the Space Shuttle Orbiter problem.

Finally, EMA was used in conjunction with range estimating to create a hybrid probabilistic methodology for evaluating the design margin of a novel concept. This hybrid methodology addresses the shortcomings in range estimating as demonstrated by the taxonomy of uncertainty. Furthermore, this methodology was demonstrated on a realistic sample problem in order to demonstrate how it can be applied to a vehicle development program. This hybrid methodology should be used to address the yet-unsolved problem of forecasting vehicle weight in order to achieve more complete and traceable forecasts.

APPENDIX A

CODE FOR OBJECT-ORIENTED IMPLEMENTATION

```
1 classdef ExecutableMatrix < handle
2     %ExecutableMatrix Highest-level object in OO EMA implementation
3     % This object is responsible for holding a list of parameters and
4     % functions which operate on an entire executable matrix of
5     % alternatives
6
7     properties
8
9         listOfParameters;
10
11        relationships; %a RelationshipMatrix object
12
13        oneWayRelationships;
14
15        combinationFactory;
16
17    end
18
19    methods
20
21        function delete(obj)
22            %destructor method— must manually clean up recursive
23            %references due to problems with Matlab garbage collection
24
25            %remove links from the list of Parameters
26            %disp('disconnecting relationships from conditions');
27
28            for i=1:size(obj.listOfParameters,1)
29
30                obj.listOfParameters{i,1}.disconnect;
```

```

31
32     end
33
34     %remove links from the relationship matrix
35     %disp('disconnecting conditions from relationships');
36     obj.relationships.disconnect;
37
38
39     end
40
41     function addParameter(obj, newParameter)
42
43         if (iscell(obj.listOfParameters) == 0)
44             %this is not initialized as a cell array therefore it is
45             % empty
46             obj.listOfParameters{1,1} = newParameter;
47
48         else
49             %cell array exists therefore the array is not empty
50             localSize = size(obj.listOfParameters, 1);
51             obj.listOfParameters{localSize + 1, 1} = newParameter;
52
53         end
54
55     end
56
57     function addOneWayRelationship(obj, newRelationship)
58
59         if (iscell(obj.oneWayRelationships) == 0)
60             %this is not initialized as a cell array therefore it is
61             % empty
62             obj.oneWayRelationships{1,1} = newRelationship;
63
64         else
65             %cell array exists therefore the array is not empty

```

```

66         localSize = size(obj.oneWayRelationships, 1);
67         obj.oneWayRelationships{localSize + 1, 1} =newRelationship;
68
69     end
70
71 end
72
73 function resetMatrix(obj)
74     % this function will iterate through all conditions resetting
75     % them to their original values before any selections are made
76
77     obj.relationships.resetMatrix;
78
79     for i=1:size(obj.listOfParameters,1)
80
81         obj.listOfParameters{i,1}.resetParameter;
82
83     end
84
85     for i=1:size(obj.oneWayRelationships,1)
86
87         obj.oneWayRelationships{i,1}.resetRelationship;
88
89     end
90
91 end
92
93 function setupRelationships(obj, relationshipFactory)
94     % This function initializes the relationships between all
95     % members of an ExecutableMatrix
96     %
97     % This will also erase all defined relationships within a
98     % defined matrix
99     %
100    %Run this function after ALL entries have been added to an

```

```

101     %ExecutableMatrix
102
103
104     obj.relationships = RelationshipMatrix(...
105         obj.listOfParameters , relationshipFactory);
106
107     %reset the executableMatrix so that all modified values are
108     %equal to the initialized values
109     obj.resetMatrix
110
111     end
112
113     function selections = selectCombination(obj, choices)
114         % This function selects a combination from the Executable Matrix
115         %
116         % choices— a 1xn array of choices where n is equal to the
117         %         number of morphological parameters. Each entry
118         %         of this array represents a condition selection from
119         %         the corresponding parameter.
120         %
121
122
123         %error check to be sure that the number of choices equals
124         % the number of parameters of the executable matrix
125
126
127
128         toReturn = obj.combinationFactory.constructNewCombination;
129
130         perm_initialization = 1:size(obj.listOfParameters,1);
131         perm_array = perms(perm_initialization);
132
133
134         selections = cell(0,0);
135         count_array = ones(1,1);

```

```

136
137     %cycle through permutation array
138     for j = 1:size(perm_array,1)
139
140         order_of_selection = perm_array(j,:);
141
142         %iterate through the list of parameters, selecting each
143         %chosen condition
144         for i = 1:size(obj.listOfParameters,1)
145
146             %find pointer to selected paramter
147             selected_index = order_of_selection(1,i);
148             parameterChosen = ...
149                 obj.listOfParameters{selected_index,1};
150
151             %gets the specific number of the condition within the
152             %parameter
153             conditionChosen = choices(selected_index,1);
154
155             %find pointer to selected condition
156             conditionToAdd = ...
157                 parameterChosen.getCondition(conditionChosen);
158
159             %select the condition in the matrix— this function
160             %triggers the relationships associated with this
161             %condition
162             conditionToAdd.select;
163
164             %renormalize parameters
165             obj.enforceAllConstraints;
166
167         end %end for i = 1:size(obj.listOfParameters,1)
168
169         %add conditions to combination selection— must do this
170         %after each selection has been chosen by the algorithm

```

```

171
172     for i = 1:size(obj.listOfParameters,1)
173
174         %find pointer to selected paramter
175         selected_index = order_of_selection(1,i);
176         parameterChosen = ...
177             obj.listOfParameters{selected_index,1};
178
179         %gets the specific number of the condition within the
180         %parameter
181         conditionChosen = choices(selected_index,1);
182
183         %find pointer to selected condition
184         conditionToAdd = ...
185             parameterChosen.getCondition(conditionChosen);
186
187         %adds a copy of the seleted condition to the return list
188         toReturn.addCondition(conditionToAdd);
189
190     end
191
192
193     %check to see if the selected combination is unique
194
195     %check to see if selections has been initialized
196     if (size(selections,1) == 0)
197         %selections has not been initalized
198
199         %add combination to the list of selections
200         selections{1,1} = toReturn;
201         %set the first part of the count array to 1
202         count_array(1,1) = 1;
203
204         %reset toReturn
205         toReturn = obj.combinationFactory.constructNewCombination;

```

```

206
207         else
208
209
210             localSize = size(selections , 1);
211             selections{localSize + 1, 1} = toReturn;
212
213             count_array(localSize + 1,1) = 1;
214
215             %reset toReturn
216             toReturn = obj.combinationFactory.constructNewCombination;
217
218         end
219
220
221         %reset the matrix to its original state
222         obj.resetMatrix;
223
224     end %end for j = 1:size(perm_array,1)
225
226
227 end
228
229 function combinations=totalCombinations(obj)
230     % This function calculates the total combinations in an
231     % executable matrix
232     %
233
234
235     %preallocate memory to calculate total combinations
236
237     simple_combs = obj.simpleCombinations;
238
239     permutations = factorial(size(obj.listOfParameters ,1));
240

```

```

241     combinations = cell((simple_combs .* permutations),1);
242
243     %iterate through each possible combination of the executable
244     % matrix
245
246
247     %this counter goes through the possible input combinations
248     %in the executable matrix
249     inputValues = ones(size(obj.listOfParameters),1),1);
250
251     %run combination function for initial case
252     temp_combs1 = selectCombination(obj, inputValues);
253     temp_combs1 = selectCompatibleCombinations(temp_combs1);
254     total = 0;
255
256     global_counter = 1;
257
258     if size(temp_combs1,1) > 0
259         %error check
260
261         for i=1:size(temp_combs1,1)
262
263             %assign index
264             combinations{global_counter,1} = temp_combs1{i,1};
265             %increment counter
266             global_counter = global_counter + 1;
267
268         end
269
270     end
271
272     %total
273
274     %reset index
275     i = 1;

```

```

276
277     while i <= size(obj.listOfParameters,1);
278
279         %increment the local counter
280         inputValues(i,1) = inputValues(i,1) + 1;
281
282         %check to see if index exceeds number of conditions
283         if inputValues(i,1) > ...
284             obj.listOfParameters{i,1}.sizeOfParameter
285
286             %reset the counter of the current index
287             inputValues(i,1) = 1;
288
289             %increment index so that the next parameter is
290             % incremented on the next pass
291             i = i + 1;
292
293     else
294
295         %This else block represents the fact that a new
296         %input value combination has been found
297
298         %get the total combinations based on the input values
299         tempCombinations= selectCombination(obj, inputValues);
300
301         %check temporary combinations to see if they are
302         %feasible
303
304         tempCombinations = ...
305             selectCompatibleCombinations(tempCombinations);
306
307         if size(tempCombinations,1) > 0
308             %error check to see if there is a list of
309             %combinations ready to add
310

```

```

311
312         for i=1:size(tempCombinations,1)
313
314             %assign index
315             combinations{global_counter,1} =...
316                 tempCombinations{i,1};
317             %increment counter
318             global_counter = global_counter + 1;
319
320         end
321
322
323         %add the number of surviving combinations to the
324         %total
325         total = total + size(tempCombinations,1);
326
327     end
328
329     %reset counter to iterate through the first
330     %parameter again
331     i = 1;
332
333     end %end if statement
334
335     end %end while i <= size(obj.listOfParameters,1)
336
337 end %end function
338
339 function total=numberOfTotalCombinations(obj)
340     % This function calculates the total number of combinations in
341     % an executable matrix
342     %
343
344
345

```

```

346     %iterate through each possible combination of the executable
347     % matrix
348
349
350     %this counter goes through the possible input combinations
351     %in the executable matrix
352     inputValues = ones(size(obj.listOfParameters,1),1);
353
354     %run combination function for initial case
355     temp_combs1 = selectCombination(obj, inputValues);
356     temp_combs1 = selectCompatibleCombinations(temp_combs1);
357     total = 0;
358
359     if size(temp_combs1,1) > 0
360         %error check
361         combinations{1,1} = temp_combs1{: ,1};
362         total = size(combinations,1);
363
364     end
365
366     %reset index
367     i = 1;
368
369     while i <= size(obj.listOfParameters,1);
370
371         %increment the local counter
372         inputValues(i,1) = inputValues(i,1) + 1;
373
374         %check to see if index exceeds number of conditions
375         if inputValues(i,1) > ...
376             obj.listOfParameters{i,1}.sizeOfParameter
377
378             %reset the counter of the current index
379             inputValues(i,1) = 1;
380

```

```

381         %increment index so that the next parameter is
382         % incremented on the next pass
383         i = i + 1;
384
385     else
386
387         %This else block represents the fact that a new
388         %input value combination has been found
389
390         %get the total combinations based on the input values
391         tempCombinations= selectCombination(obj, inputValues);
392
393         %check temporary combinations to see if they are
394         %feasible
395
396         tempCombinations = ...
397             selectCompatibleCombinations(tempCombinations);
398
399
400         %add the number of surviving combinations to the
401         %total
402         total = total + size(tempCombinations,1);
403
404
405
406         %reset counter to iterate through the first
407         %parameter again
408         i = 1;
409
410     end %end if statement
411
412     end %end while i <= size(obj.listOfParameters,1)
413
414 end
415

```

```

416 function enforceAllConstraints(obj)
417     % this function calls the enforce constraints function for each
418     % parameter in the executable matrix
419     for i=1:size(obj.listOfParameters,1)
420
421         obj.listOfParameters{i,1}.enforceConstraints;
422
423     end
424
425 end
426
427 function initializeConditions(obj)
428     % this function calls the initializeParameter function for each
429     % parameter in the executable matrix
430     for i=1:size(obj.listOfParameters,1)
431
432         obj.listOfParameters{i,1}.initializeParameter;
433
434     end
435 end
436
437 function unInitializeConditions(obj)
438     % this function calls the initializeParameter function for each
439     % parameter in the executable matrix
440     for i=1:size(obj.listOfParameters,1)
441
442         obj.listOfParameters{i,1}.unInitializeParameter;
443
444     end
445 end
446
447
448 function total = simpleCombinations(obj)
449     %this function calculates the total number of simple
450     %combinations for the morphological field

```

```

451
452     total = 1;
453
454     for i=1:size(obj.listOfParameters,1)
455
456         total = total .* obj.listOfParameters{i,1}.sizeOfParameter;
457
458     end
459
460 end
461
462 function SQ = solutionQuotient(obj)
463
464     simpleCombinations = obj.simpleCombinations;
465
466     feasibleCombinations = obj.numberofTotalCombinations;
467
468     SQ = feasibleCombinations ./ simpleCombinations;
469
470
471 end
472
473 end %end public methods
474
475 methods (Abstract)
476
477     table=monteCarlo(obj,baseline , numRuns)
478
479
480 end
481
482 end
483
484
485 %Helper sub-functions

```

```

486
487 function updatedList = selectCompatibleCombinations(listOfCombinations)
488 %this function is a helper function for totalCombinations— its job is
489 %to return only members of the input list which are compatible (have a
490 %probability > 0 )
491 %
492 % listOfCombinations: a 1xn cell array of Combination objects— this should
493 % represent the combinations which were yielded by selectCombination
494
495 sizeOfUpdatedList = 0;
496
497 updatedList = cell(0,0);
498
499 for i=1:size(listOfCombinations, 1)
500
501     if listOfCombinations{i,1}.isFeasible == 1
502         %this means that a combination is feasible, add it to the
503         %updated list
504
505         %increment counter
506         sizeOfUpdatedList = sizeOfUpdatedList + 1;
507
508         %actually add the combination to the list
509         updatedList{sizeOfUpdatedList,1} = listOfCombinations{i,1};
510
511     end%end if statement
512
513 end %end for loop
514
515
516 end

```

```

1 classdef Parameter < handle
2     %PARAMETER Represents a single parameter of a morphological field
3
4

```

```

5   properties
6
7       name;
8
9       listOfConditions;
10
11      sizeOfParameter = 0;
12
13      selected = 0;
14
15  end
16
17  methods
18
19      function addCondition(obj, newAlternative)
20
21          newAlternative.parentParameter = obj;
22
23          if (iscell(obj.listOfConditions) == 0)
24              %this is not initialized as a cell array therefore it is
25              % empty
26              obj.listOfConditions{1,1} = newAlternative;
27
28              %update size of category
29              obj.sizeOfParameter = 1;
30          else
31              %cell array exists therefore the array is not empty
32              localSize = size(obj.listOfConditions, 1);
33              obj.listOfConditions{localSize + 1, 1} = newAlternative;
34
35              %update size of category
36              obj.sizeOfParameter = obj.sizeOfParameter + 1;
37          end
38
39  end

```

```

40
41 function condition = getCondition(obj, index)
42     %returns a pointer to the specified alternative in a category
43
44     if (index <= 0) && (index > obj.sizeOfParameter)
45         condition = NaN;
46     else
47         condition = obj.listOfConditions{index , 1};
48     end
49
50 end
51
52
53 function initializeParameter(obj)
54     %initializes each of the conditions within a parameter
55
56     %iterate through each set condition in the parameter
57     for i=1:size(obj.listOfConditions , 1)
58         obj.listOfConditions{i,1}.initializeCondition;
59     end
60
61     %make sure that the parameter is well-conditioned after
62     %initialization
63     obj.enforceConstraints();
64
65 end
66
67 function unInitializeParameter(obj)
68     %un-initializes each of the conditions within a parameter
69
70     %iterate through each set condition in the parameter
71     for i=1:size(obj.listOfConditions , 1)
72         obj.listOfConditions{i,1}.unInitializeCondition;
73     end
74

```

```

75     end
76
77
78     function resetParameter(obj)
79         %resets each of the conditions within a parameter
80
81         obj.selected = 0;
82
83         %iterate through each set condition in the parameter
84         for i=1:size(obj.listOfConditions , 1)
85             obj.listOfConditions{i,1}.resetCondition
86         end
87
88     end
89
90     function disconnect(obj)
91         %resets each of the conditions within a parameter
92
93         %iterate through each set condition in the parameter
94         for i=1:size(obj.listOfConditions , 1)
95             obj.listOfConditions{i,1}.disconnect
96         end
97
98     end
99
100
101 end
102
103 methods (Abstract)
104
105     enforceConstraints(obj)
106
107 end
108
109 end

```

```

1 classdef Condition < handle
2     %CONDITION Represents a single Condition
3     % Contains the information necessary for a single Condition
4
5     properties
6
7         %standard properties which do not change once set
8         name;
9
10        %a list of the relationships that this condition has with other
11        % conditions in the relationship matrix
12        listOfRelationships;
13
14
15
16
17        modified; % this is a boolean flag set to true if a modification
18                % operator has occurred to this— useful for resetting the
19                % state of the condition
20
21
22    end
23
24
25
26    methods
27
28        function delete(obj)
29            %destructor function
30            obj.disconnect;
31        end
32
33        function disconnect(obj)
34            %must delete references to relationships to avoid weird memory
35            %issue when clearing data

```

```

36
37     obj.listOfRelationships = [];
38
39 end
40
41 function select(obj)
42     %this function "selects" a condition — this requires that all
43     %relationships that are part of this condition be activated
44     %
45
46     for i=1:size(obj.listOfRelationships,1)
47         %load member of the list
48         tempRelationship = obj.listOfRelationships{i,1};
49
50         %execute the relationship
51         tempRelationship.activateRelationship(obj);
52
53     end %end for i=1:size(obj.listOfRelationships,1)
54
55
56 end
57
58 function resetCondition(obj)
59     %this resets the modified probability and modified effects to
60     %be equal to the original input values
61     %
62     % this should be run after a selection run
63
64     obj.modified = false;
65
66
67 end
68
69 function addRelationship(obj, newRelationship)
70

```

```

71         if (iscell(obj.listOfRelationships) == 0)
72             %this is not initialized as a cell array therefore it is
73             % empty
74             obj.listOfRelationships{1,1} = newRelationship;
75
76         else
77             %cell array exists therefore the array is not empty
78             localSize = size(obj.listOfRelationships, 1);
79             obj.listOfRelationships{localSize + 1, 1} =newRelationship;
80
81         end
82
83     end
84
85     function equality = isEqual(obj, condition2)
86
87         %string compare the names
88         if strcmp(obj.name, condition2.name) == 0
89             equality = 0;
90         end
91
92     end
93 end
94 end

```

```

1 classdef Combination < handle
2     %COMBINATION Contains the information of selected conditions
3     %
4
5     properties
6
7         %This list must always be sorted in order of the executable matrix
8         %parameters with a single condition from each parameter
9         listOfSelectedConditions;
10
11

```

```

12     end
13
14     methods
15
16         function obj = Combination(varargin)
17
18             if size(varargin,1) > 0
19                 obj.listOfSelectedConditions = cell(varargin{1,1},1);
20             else
21                 obj.listOfSelectedConditions = cell(0,0);
22             end
23
24         end
25
26     end
27
28     methods (Abstract)
29
30         addCondition(obj, conditionToAdd, varargin)
31
32         equality = isEqual(obj, combination2)
33
34         feasibility = isFeasible(obj)
35
36     end
37
38 end

```

```

1 classdef CombinationFactory
2     %COMBINATIONFACTORY Factory Class for Combination Factory design
3     %pattern
4     %
5
6     properties
7     end
8

```

```

9     methods (Abstract , Static)
10
11         combination = constructNewCombination(sizeOfParameterList)
12
13     end
14
15 end

```

```

1 classdef RelationshipMatrix
2     %RELATIONSHIPMATRIX Representation of the relationships matrix
3     % Contains the actual realtionships as well as necessary function to
4     % operate and construct the Relationship Matrix
5
6     properties
7
8         matrix_of_relationships % the matrix of relationships
9
10        relationship_Factory % factory class for creating specific types of
11                               % relationships
12
13    end
14
15    methods
16
17        function obj = RelationshipMatrix(listOfParameters , ...
18            relationship_Factory)
19            % This function initializes the relationships between all
20            % members of an ExecutableMatrix
21            %
22            % This will also erase all defined relationships within a
23            % defined matrix
24            %
25            %Run this function after ALL entries have been added to an
26            %ExecutableMatrix
27
28            %set relationship_Factory

```

```

29     obj.relationship_Factory = relationship_Factory;
30
31     %reset relationshipMatrix
32     obj.matrix_of_relationships = ...
33         relationship_Factory.constructNewRelationship;
34
35     %determine the total number of categories in the matrix
36     number_of_categories = size(listOfParameters, 1);
37
38     for i=2: number_of_categories
39         %outer loop to define 1:1 relationship matrix
40         %start with the 2nd row and iterate until the final row
41
42         current_category = listOfParameters{i,1};
43         for ii=1:current_category.sizeOfParameter
44             %iterate through each condition of the current category
45
46             condition = current_category.getCondition(ii);
47
48             for j=1:(i-1)
49                 %iterate through the other side creating 1:1
50                 %relationships for each pair of conditions
51
52                 current_category_j = listOfParameters{j,1};
53
54                 for jj=(1:current_category_j.sizeOfParameter)
55                     %cycle through each member of the new category
56
57                     condition_j=current_category_j.getCondition(jj);
58
59                     %Finally, create a new relationship
60                     toadd = ....
61                     relationship_Factory.constructNewRelationship(
62                         condition, condition_j);

```

```

63      %add relationship to the two conditions
64
65      %add relationship to original condition
66      condition.addRelationship(toadd);
67      %add relationship to second condition
68      condition_j.addRelationship(toadd);
69
70      %add the relationship to the relationshipMatrix
71
72      %calculate necessary row
73      row_index = 0;
74
75      %add up total number of conditions in categories
76      % already covered by the algorithm
77      for i_x = 2:(i-1)
78          row_index = row_index + ...
79              listOfParameters{i_x,1}.sizeOfParameter;
80      end
81
82      % add the current index number to the total
83
84      row_index = row_index + ii;
85      %calculate necessary column
86      column_index = 0;
87
88      %add up total number of conditions in the column
89      for j_y = 1:(j-1)
90          column_index = column_index + ...
91              listOfParameters{j_y,1}.sizeOfParameter;
92
93      end
94
95      %add the curent index number to total
96      column_index = column_index + jj;
97

```

```

98         %add relationship to the matrix
99         obj.matrix_of_relationships(row_index ,...
100             column_index)= toadd;
101
102     end %end for jj=(1:current_category_sub.size)
103
104     end %j=1:(i-1)
105
106     end %end for ii=1:current_category.size
107
108     end %end i=2: number_of_categories
109
110 end
111
112
113 function resetMatrix(obj)
114
115     %cycle through each row
116     for i=1:size(obj.matrix_of_relationships,1)
117
118         %cycle through the non-empty parts of the relationship
119         %matrix
120         for j = 1:size(obj.matrix_of_relationships,2)
121
122
123             %if the counter has reached an empty field , break the
124             %loop; it was implemented this way to avoid the case
125             %where j > the dimensions of the relationship matrix
126             if obj.matrix_of_relationships(i,j).empty == 1
127                 break
128             end
129
130             obj.matrix_of_relationships(i,j).resetRelationship;
131
132     end

```

```

133
134         end
135
136
137     end
138
139     function disconnect(obj)
140         %disconnects all references
141
142         for i=1:size(obj.matrix_of_relationships,1)
143
144             for j = 1:size(obj.matrix_of_relationships,2)
145
146                 obj.matrix_of_relationships(i,j).disconnect;
147
148             end
149
150         end
151
152     end
153
154 end
155
156 end

```

```

1 classdef RelationshipFactory
2     %RELATIONSHIPFACTORY Abstract Implementation of Relationship Factory
3     %
4
5     properties
6
7     end
8
9     methods (Abstract, Static)
10
11         relationship = constructNewRelationship(end1, end2)

```

```
12
13     end
14
15 end
```

```
1 classdef Relationship < handle
2     %RELATIONSHIP This represents a relationship between two conditions
3     %           of an executable matrix of alternatives
4     %
5
6     properties
7
8
9         activated = 0; %boolean to make sure that each relationship is
10            %only activated once
11
12         empty = 1; %this is a boolean that says that this is an empty
13            %relationship. This is needed because matlab does not
14            %like empty cells in an array.
15
16
17     end
18
19     properties (SetAccess = private , GetAccess = public)
20         one;
21         two;
22
23     end
24
25     methods
26
27         function disconnect(obj)
28             %disconnect the condition linkages for memory management
29             %purposes
30             obj.one = [];
31             obj.two = [];
```

```

32     end
33
34     function delete(obj)
35         %destructor function
36         %must delete references to conditions to avoid weird memory
37         %issue when clearing data
38
39         obj.disconnect;
40
41     end
42
43     function obj = Relationship(varargin)
44
45         if size(varargin,1) > 0
46             obj.one = varargin{1,1};
47             obj.two = varargin{1,2};
48             obj.empty = 0; %declares that this is not empty
49         end
50     end
51
52     function resetRelationship(obj)
53
54         obj.activated = 0;
55
56     end
57
58
59 end
60
61 methods (Abstract)
62
63     activateRelationship(obj, activatingCondition)
64
65
66 end

```

67

68 **end**

```
1 classdef ForecastingExecutableMatrix < ExecutableMatrix
2     %FORECASTINGEXECUTABLEMATRIX Specific implementation an
3     %ExecutableMatrix for forecasting problems
4     % Contains functions for calculating an expected value, conducting a
5     % Monte Carlo simulation, and the direct calculation of the CDF
6
7 properties
8     end
9
10    methods
11
12
13        function value=expectedValue(obj, baseline)
14
15            combinations = obj.totalCombinations;
16            value = 0;
17
18            %reapportion probabilities
19
20            prob_sum = 0;
21
22            for i=1:size(combinations,1)
23
24                prob_sum = prob_sum + combinations{i,1}.totalProbability;
25
26            end
27
28
29            for i=1:size(combinations,1)
30
31                combinations{i,1}.overRiddenProbability = ...
32                    combinations{i,1}.totalProbability ./ prob_sum;
33
```

```

34         end
35
36
37         for i=1:size(combinations,1)
38
39             [add mult] = combinations{i,1}.totalEffect;
40
41             local_effect = (baseline + add) .* mult;
42
43             value = value + ...
44                 combinations{i,1}.overRiddenProbability .* local_effect;
45
46         end
47
48
49     end
50
51     function table=monteCarlo(obj, baseline , numRuns)
52
53         table = ones(numRuns,2);
54
55
56
57         %for loop through the number of runs
58         for i=1:numRuns
59             %need to randomize the selection of parameters because the
60             %order in which parameters are chosen matters
61
62             tankage_value = [0 1];
63             structures_value = [0 1];
64             tps_value = [0 1];
65             propulsion_value = [0 1];
66
67
68             %determine potential indexes for selection

```

```

69      indices = 1:size(obj.listOfParameters,1);
70      selected_indices = ones(size(obj.listOfParameters,1),1);
71
72      %create a new variable for the value for an individual
73      %Monte Carlo run
74      all_value = [0 1];
75      %for loop through each category
76      for j=1:size(obj.listOfParameters, 1)
77
78          %roll dice to see which index is chosen
79          random_index = ...
80              1 + floor(mod(10 * rand(1,1), size(indices,2)));
81
82          index_value = indices(random_index);
83
84          %execute the Monte Carlo sim on the parameter
85          [selection_index] = ...
86              obj.listOfParameters{index_value,1}.subMonteCarlo;
87
88          %enforce constraints
89          obj.enforceAllConstraints;
90
91          selected_indices(index_value,1) = selection_index;
92
93
94          %remove index from the list of indices
95          indices(random_index) = [];
96
97
98      end%end for j=1:size(obj.listOfParameters, 1)
99
100     %extract selected conditions
101
102     for j=1:size(obj.listOfParameters, 1)
103

```

```

104      %load local parameter
105      localParameter = obj.listOfParameters{j,1};
106
107      %load local condition
108      local_condition = ...
109          localParameter.getCondition(selected_indices(j,1));
110
111      %extract condition information
112      subsystem = localParameter.subsystemAssignment;
113      add = local_condition.modifiedAdditiveEffect;
114      mult = local_condition.modifiedMultiplicativeEffect;
115
116      switch(subsystem)
117          case Subsystems.tankage
118              tankage_value(1,1) = tankage_value(1,1) + add;
119              tankage_value(1,2) = tankage_value(1,2) .* mult;
120          case Subsystems.structures
121              structures_value(1,1) = ...
122                  structures_value(1,1) + add;
123              structures_value(1,2) = ...
124                  structures_value(1,2) .* mult;
125          case Subsystems.tps
126              tps_value(1,1) = tps_value(1,1) + add;
127              tps_value(1,2) = tps_value(1,2) .* mult;
128          case Subsystems.propulsion
129              propulsion_value(1,1) = ...
130                  propulsion_value(1,1) + add;
131              propulsion_value(1,2) = ...
132                  propulsion_value(1,2) .* mult;
133          case Subsystems.all
134              all_value(1,1) = all_value(1,1) + add;
135              all_value(1,2) = all_value(1,2) .* mult;
136
137      end % end switch
138

```

```

139         end
140
141         %table(i,1) = tankage_value(1,1);
142         %table(i,2) = tankage_value(1,2);
143         %table(i,3) = structures_value(1,1);
144         %table(i,4) = structures_value(1,2);
145         %table(i,5) = tps_value(1,1);
146         %table(i,6) = tps_value(1,2);
147         %table(i,7) = propulsion_value(1,1);
148         %table(i,8) = propulsion_value(1,2);
149
150         table(i,1) = all_value(1,1);
151         table(i,2) = all_value(1,2);
152
153         %reset the matrix
154         obj.resetMatrix;
155
156     end%end for i=1:numRuns
157
158     %apply baseline value
159     table = (baseline + table(:,1)) .* table(:,2);
160
161 end %end function monteCarlo
162
163
164
165 function [f x] = generateCDF(obj, baseline)
166     %this function will generate a CDF table by generating every
167     %possible combination of the matrix
168     %
169     % baseline— the baseline value for which multiplicative
170     % relationships will be applied
171
172     %get all combinations
173     combinations = obj.totalCombinations;

```

```

174
175
176
177     %reapportion probabilities
178
179     prob_sum = 0;
180
181     for i=1:size(combinations,1)
182
183         prob_sum = prob_sum + combinations{i,1}.totalProbability;
184
185     end
186
187
188     for i=1:size(combinations,1)
189
190         combinations{i,1}.overRiddenProbability = ...
191             combinations{i,1}.totalProbability ./ prob_sum;
192
193     end
194
195
196     %preallocate memory
197     cdfTable = ones(size(combinations,1),2);
198
199     %load information into a table
200     for i=1:size(combinations,1)
201
202         [additiveEffect multiplicativeEffect] =...
203             combinations{i,1}.totalEffect;
204
205         effect = (baseline + additiveEffect) .* ...
206             multiplicativeEffect;
207
208         cdfTable(i,1) = effect;

```

```

209         cdfTable(i,2) = combinations{i,1}.overRiddenProbability;
210
211
212     end
213
214     %sort by probability; ~ ignores output
215     [~, I] = sort(cdfTable(:,1));
216
217     cdfTable = cdfTable(I,:);
218
219     %set values
220     x = cdfTable(:,1);
221
222     %repeat first value for the 0 chance probability to complete CDF
223     x = [x(1,1) ; x(:,1)];
224
225
226     %generate continuous distribution
227     f(1,1) = 0;
228
229     for i=2:(size(cdfTable,1) + 1)
230
231         f(i,1) = f(i-1,1) + cdfTable(i-1,2);
232
233     end
234
235 end
236
237 end
238
239 end

```

```

1 classdef ForecastingParameter < Parameter
2     %FORECASTINGPARAMETER Specific implementation of a parameter for a
3     %weight forecasting problem
4     %

```

```

5
6   properties
7
8       subsystemAssignment =Subsystems.all; % Subsystem to which this
           parameter belongs to
9
10  end
11
12  methods
13
14      function enforceConstraints(obj)
15          %this function will renormalize the modified effects for this
16          %parameter so that the sum total of the modified effects equals
17          %1
18
19          %preallocate memory
20          probabilities = ones(obj.sizeOfParameter , 1);
21
22          %load the probabilities of the conditions into the vector
23          for i=1:obj.sizeOfParameter
24              condition = obj.getCondition(i);
25              probabilities(i,1) = condition.modifiedProbability;
26          end
27
28          %normalize probabilities
29          vector_sum = sum(probabilities);
30          probabilities = probabilities/vector_sum;
31
32
33          %load the probabilities of the vector back into the conditions
34          for i=1:obj.sizeOfParameter
35              condition = obj.getCondition(i);
36              condition.modifiedProbability = probabilities(i,1);
37          end
38

```

```

39     end
40
41     function [selection_index name]=subMonteCarlo(obj)
42         % [selection_index subsystem name]=subMonteCarlo(obj)
43         %subsystem = obj.subsystemAssignment;
44
45
46         %draw random value
47         randomValue = rand(1,1);
48
49
50         %determine which alternative to pick
51         cumulative_probability = 0; %variable to act as a datum for
52         %probabilities
53         for i=1:size(obj.listOfConditions, 1)
54
55             %update cumulative probability
56             cumulative_probability = cumulative_probability + ...
57                 obj.listOfConditions{i,1}.modifiedProbability;
58
59             if randomValue < cumulative_probability
60                 %we have found the correct alternative
61
62                 %select the condition
63                 obj.listOfConditions{i,1}.select;
64
65
66                 %return the correct value
67
68                 selection_index = i;
69
70                 name = ...
71                     obj.listOfConditions{i,1}.name;
72
73                 %break the for loop

```

```

74         break
75     end %end if statement
76
77     end %end for loop
78
79     end %end function subMonteCarlo
80
81
82 end
83
84 end

```

```

1 classdef ForecastingCondition < Condition
2     %FORECASTINGCONDITION Specific implementation of a Condition for a
3     %weight forecasting problem
4     % This includes problem-specific properties and methods
5
6     properties
7
8         %standard properties which do not change once set
9         probability = 1;
10        additiveEffect = 0;
11        multiplicativeEffect = 1;
12
13        % modified parameters based on relationships from other
14        % (previously selected) conditions
15        modifiedProbability = 1;
16        modifiedAdditiveEffect = 0;
17        modifiedMultiplicativeEffect = 1;
18
19    end
20
21    methods
22
23
24        function equality = isEqual(obj, condition2)

```

```

25      %two conditions will be deemed equal if they have the same
26      %effect and probability
27      %
28      % return value:
29      % 0 means that two conditions are not equal
30      % 1 means that both conditions are equal
31      %
32      %Modified effect and modified probability do not matter since
33      %these are transitional states during the selection of a single
34      %combination. At the conclusion of a selection process, new
35      %condition objects are made to be part of combinations, and the
36      %original executable matrix is reset
37
38      %start off with the assumption that it will be equal, any
39      %inequality will change this flag to 0
40      equality = 1;
41
42      if obj.probability ~= condition2.probability
43          equality = 0;
44      end
45
46      if obj.additiveEffect ~= condition2.additiveEffect
47          equality = 0;
48      end
49
50      if obj.multiplicativeEffect ~= condition2.multiplicativeEffect
51          equality = 0;
52      end
53
54      %string compare the names
55      if strcmp(obj.name, condition2.name) == 0
56          equality = 0;
57      end
58
59      end

```

```

60
61
62     function resetCondition(obj)
63         %this resets the modified probability and modified effects to
64         %be equal to the original input values
65         %
66         % this should be run after a selection run
67
68         obj.modifiedProbability = obj.probability;
69         obj.modifiedAdditiveEffect = obj.additiveEffect;
70         obj.modifiedMultiplicativeEffect = obj.multiplicativeEffect;
71         obj.modified = false;
72
73
74     end
75
76
77 end
78
79 end

```

```

1  classdef ForecastingCombination < Combination
2      %FORECASTINGCOMBINATION Specific implementation of a combination for a
3      %weight forecasting problem
4      %
5
6      properties
7
8          overRiddenProbability;
9
10     end
11
12     methods
13
14         function obj = ForecastingCombination(varargin)
15

```

```

16
17     %call the superclass constructor
18     % n.b.: varargin{:} creates a comma-separated list from the
19     % cell array that is varargin
20     obj = obj@Combination(varargin{:});
21 end
22
23
24 function addCondition(obj, conditionToAdd, varargin)
25
26     numvarargs = length(varargin);
27
28     if numvarargs > 0
29
30         position = varargin{1};
31
32         %define a condition class to add to the combination
33         newCondition = ForecastingCondition;
34
35         %copy the data from the condition to a new object
36         newCondition.name = conditionToAdd.name;
37         newCondition.probability = ...
38             conditionToAdd.modifiedProbability;
39         newCondition.additiveEffect = ...
40             conditionToAdd.modifiedAdditiveEffect;
41         newCondition.multiplicativeEffect = ...
42             conditionToAdd.modifiedMultiplicativeEffect;
43
44         obj.listOfSelectedConditions{position,1} = ...
45             newCondition;
46
47     else
48
49         %define a condition class to add to the combination
50         newCondition = ForecastingCondition;

```

```

51
52     %copy the data from the condition to a new object
53     newCondition.name = conditionToAdd.name;
54     newCondition.probability = ...
55         conditionToAdd.modifiedProbability;
56     newCondition.additiveEffect = ...
57         conditionToAdd.modifiedAdditiveEffect;
58     newCondition.multiplicativeEffect = ...
59         conditionToAdd.modifiedMultiplicativeEffect;
60
61     if (iscell(obj.listOfSelectedConditions) == 0)
62         %this is not initialized as a cell array therefore it is
63         % empty
64         obj.listOfSelectedConditions{1,1} = newCondition;
65
66     else
67         %cell array exists therefore the array is not empty
68         localSize = size(obj.listOfSelectedConditions, 1);
69         obj.listOfSelectedConditions{localSize + 1, 1} =...
70             newCondition;
71
72     end
73 end
74 end
75
76
77 function value = expectedValue(obj)
78     %old implementation, deprecated
79     value = 0;
80
81     %iterate through each condition of the category
82     for i=1:size(obj.listOfSelectedConditions,1)
83         localProb = obj.listOfSelectedConditions{i,1}.probability;
84         localEffect = obj.listOfSelectedConditions{i,1}.effect;
85

```

```

86         value = value + localProb .* localEffect;
87
88     end
89
90 end
91
92 function prob = totalProbability(obj)
93     % This function calculates the total probability of the
94     % combination being selected— this is the product of each
95     % condition's individual probability
96     %
97
98     prob = 1;
99
100    %iterate through each condition of the category
101    for i=1:size(obj.listOfSelectedConditions,1)
102        localProb = obj.listOfSelectedConditions{i,1}.probability;
103
104        prob = prob .* localProb;
105
106    end
107
108 end
109
110 function [additiveEffect multiplicativeEffect] = totalEffect(obj)
111     % This function calculates the total effect of the
112     % combination being selected— this is the sum of each
113     % condition's individual effect
114     %
115
116     additiveEffect = 0;
117     multiplicativeEffect = 1;
118
119     %iterate through each condition of the category
120     for i=1:size(obj.listOfSelectedConditions,1)

```

```

121         localAEffort = ...
122             obj.listOfSelectedConditions{i,1}.additiveEffect;
123
124         localMEffort = ...
125             obj.listOfSelectedConditions{i,1}.multiplicativeEffect;
126
127         additiveEffect = additiveEffect + localAEffort;
128
129         multiplicativeEffect = multiplicativeEffect .* ...
130             localMEffort;
131     end
132
133 end
134
135
136
137 function equality = isEqual(obj, combination2)
138
139     equality = 1;
140
141     for i=1:size(obj.listOfSelectedConditions, 1)
142
143         equality = equality * ...
144             obj.listOfSelectedConditions{i,1}.isEqual(...
145                 combination2.listOfSelectedConditions{i,1});
146
147     end
148
149 end
150
151 function feasibility = isFeasible(obj)
152     % this function will check to see if this combination is
153     % feasible: whether or not it has a probability greater than 0
154
155     feasibility = 1;

```

```

156
157     for i=1:size(obj.listOfSelectedConditions,1)
158
159         %if a single member of the combination has a probability of
160         %0 then the combination is infeasible
161         if obj.listOfSelectedConditions{i,1}.probability == 0
162             feasibility = 0;
163         end
164
165     end%end for loop
166
167 end
168
169
170 end
171
172 end

```

```

1 classdef ForecastingCombinationFactory < CombinationFactory
2     %FORECASTINGCOMBINATIONFACTORY Concrete Instantiation of a Combination
3     %Factory for Forecasting Problems
4     %
5
6     properties
7     end
8
9     methods (Static)
10
11         function combination = constructNewCombination(sizeOfParameterList)
12
13             if nargin > 0
14
15                 nargin
16
17                 combination = ForecastingCombination(sizeOfParameterList);
18

```

```

19         else
20             combination = ForecastingCombination();
21         end
22
23
24     end
25
26 end
27
28 end

```

```

1 classdef ForecastingRelationship < Relationship
2     %FORECASTINGRELATIONSHIP Specific implementation of a Relationship for
3     % a weight forecasting problem
4     %
5
6     properties
7         probability = 1; %defaulted to no change in probability
8         additiveRelationship = 0; %defaulted to no change in effect
9         multiplicativeRelationship = 1; %defaulted to no change in effect
10
11         additive_toMult_Relationship = 0; %decaulted to no change in effect
12         multiplicative_toMult_Relationship = 1; %decaulted to no change in
13         effect
14
15     end
16
17     methods
18
19         function obj = ForecastingRelationship(varargin)
20             % This constructor can be used to instantiate a blank
21             % relationship (i.e. with no connections) or one with two
22             % conditions for ends
23
24             %call the superclass constructor

```

```

25     % n.b.: varargin{:} creates a comma-separated list from the
26     % cell array that is varargin
27     obj = obj@Relationship(varargin{:});
28
29
30     end
31
32     function activateRelationship(obj, activatingCondition)
33         %this function will activate the relationship from one end of
34         %the link
35         %
36         %activatingCondition is a Condition object of the Condition
37         %which has been selected; this function will change the
38         %modified properties of the condition on the opposite end of
39         %the activatingCondition
40
41         %declare the variable toActivate outside of the if/then scope
42         toActivate = NaN;
43
44         %detect which handle to activate— this operates on checking
45         %the equality of the handle
46
47         activated1 = 0;
48         activated2 = 0;
49
50         if obj.one == activatingCondition
51             toActivate = obj.two;
52
53             activated2 = 1;
54         elseif obj.two == activatingCondition
55             toActivate = obj.one;
56
57             activated1 = 1;
58         end
59     end

```

```

60
61     if (activated1 + activated2) <= 0
62
63         disp('Error in relationship definiton')
64         str=[activatingCondition.name, ' selected; ', ...
65             obj.one.name, ' and ', obj.two.name, ...
66             ' are the conditions in the relationships']
67         disp(str)
68
69     end
70
71
72
73
74     %%% Activate Probability Relationship %%%
75     if ~toActivate.parentParameter.selected
76         %only activate if target's parent parameter has not
77         %previously been selected in the algorithm
78
79
80         %multiplicative operation on the probability
81         toActivate.modifiedProbability = ...
82             toActivate.modifiedProbability .* obj.probability;
83     end
84
85     %%% Activate Relationships Affecting Additive Effects %%%
86
87     %in order of operations, multiplicative relationship is applied
88     %first
89
90     %The multiplicative effect is applied to both conditions
91     if activated1
92         obj.one.modifiedAdditiveEffect = ...
93             obj.one.modifiedAdditiveEffect .* ...
94             obj.multiplicativeRelationship;

```

```

95     end
96
97     if activated2
98         obj.two.modifiedAdditiveEffect = ...
99         obj.two.modifiedAdditiveEffect .* ...
100        obj.multiplicativeRelationship;
101     end
102
103     % Additive relationship is only applied once
104     if ~obj.activated
105         toActivate.modifiedAdditiveEffect = ...
106         toActivate.modifiedAdditiveEffect +...
107         obj.additiveRelationship;
108     end
109
110     %%% Activate Relationships Affecting Multiplicative Effects %%%
111
112     %multiplicative relationship is only applied once
113     if ~obj.activated
114         toActivate.modifiedMultiplicativeEffect = ...
115         toActivate.modifiedMultiplicativeEffect .*...
116         obj.multiplicative_toMult_Relationship;
117     end
118
119     %The additive effect is applied to both conditions
120     if activated1
121         obj.one.modifiedMultiplicativeEffect = ...
122         obj.one.modifiedMultiplicativeEffect +...
123         obj.additive_toMult_Relationship;
124     end
125
126     if activated2
127         obj.two.modifiedMultiplicativeEffect = ...
128         obj.two.modifiedMultiplicativeEffect +...
129         obj.additive_toMult_Relationship;

```

```

130         end
131
132
133         %%% Update the Relationship 's State to Activated %%%
134         obj.activated = true;
135     end
136
137     function RE = calculateRelationshipEffect(obj, baseline)
138
139         %declare return value
140         RE = RelationshipEffect;
141
142         %check to see if this is an empty relationship
143         if obj.empty
144             RE.empty = 1;
145         else
146             RE.empty = 0;
147
148             %calculate probability — each condition times the
149             %relationship
150
151             RE.probability = obj.one.probability .* ...
152                 obj.two.probability .* obj.probability;
153
154             %calculate effect
155
156             %multiplicative effect
157
158             totalME = (obj.one.multiplicativeEffect .* ...
159                 obj.two.multiplicativeEffect .* ...
160                 obj.multiplicative_toMult_Relationship) + ...
161                 obj.additive_toMult_Relationship;
162
163             %additive effect
164

```

```

165         totalAE = obj.multiplicativeRelationship .* ...
166             (obj.one.additiveEffect + ...
167             obj.two.additiveEffect) + obj.additiveRelationship;
168
169         RE.effect = totalME .* (baseline + totalAE);
170
171         %calculate the expected value
172
173         RE.expectedValue = RE.probability .* RE.effect;
174
175         %assign the names
176
177         RE.name1 = obj.one.name;
178         RE.name2 = obj.one.name;
179
180
181         end
182
183     end
184
185
186 end
187
188 end

```

```

1 classdef ForecastingRelationshipFactory < RelationshipFactory
2     %FORECASTINGRELATIONSHIPFACTORY Specific implementation of a factory
3     %design pattern to generate ForecastingRelationships
4     %
5     properties
6     end
7
8     methods (Static)
9
10         function relationship = constructNewRelationship(end1, end2)
11

```

```
12         if nargin >0
13             relationship = ForecastingRelationship(end1, end2);
14
15         else
16             relationship = ForecastingRelationship();
17         end
18
19
20
21     end
22
23 end
24
25 end
```

REFERENCES

- [1] AACE INTERNATIONAL, “Contingency estimating - general principles,” 2008.
- [2] AACE INTERNATIONAL, “Risk analysis and contingency determination using range estimating,” 2008.
- [3] AACE INTERNATIONAL, “Risk analysis and contingency determination using expected value,” 2009.
- [4] AACE INTERNATIONAL, “Risk analysis and contingency determination using parametric estimating,” 2011.
- [5] AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS, “Recommended practice for mass properties control for satellites, missiles, and launch vehicles,” 1999.
- [6] AMERICAN INSTITUTE OF AERONAUTICS AND ASTRONAUTICS, “Mass properties control for space systems,” 2006.
- [7] AYYUB, B. and CHAO, R., “Uncertainty modeling in civil engineering with structural reliability applications,” in *Uncertainty modeling and analysis in civil engineering* (AYYUB, B., ed.), CRC, 1998.
- [8] BARRAZA, G., “Probabilistic estimation and allocation of project time contingency,” *Journal of Construction Engineering and Management*, vol. 137, p. 259, 2011.
- [9] BARRAZA, G. and BUENO, R., “Cost contingency management,” *Journal of Management in Engineering*, vol. 23, p. 140, 2007.
- [10] BATE, R., MUELLER, D., and WHITE, J., *Fundamentals of Astrodynamics*. 1971.
- [11] BERGIN, C., “Orion’s plea to ares i: Stop adversely hindering our design process.” <http://www.nasaspaceflight.com/2008/09/orions-plea-to-ares-i-stop-adversely-hindering-our-design-process/>, 2008.
- [12] BIGGS, R., LOVE, M., and PENDLETON, E., “An integrated airframe experiment for future responsive access to space applications,” in *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2009.
- [13] BIGGS, R., MCCANDLESS, M., CLARK, B., COCHRAN, R., PENDLETON, E., GRIFIN, K., and ZINK, S., “Structural efficiency of integrated composite structures for future space launch vehicle airframe applications,” in *AIAA SPACE 2011 Conference & Exposition*, American Institute of Aeronautics and Astronautics, 2011.
- [14] BILSTEIN, R., *Stages to Saturn: A Technological History of the Apollo/Saturn Launch Vehicle*. DIANE Publishing, 1999.
- [15] BILTGEN, P. T., *A methodology for capability-based technology evaluation for systems-of-systems*. PhD thesis, Georgia Institute of Technology, 2007.

- [16] BITTEN, R., FREANER, C., and EMMONS, D., “Optimism in early conceptual designs and its effect on cost and schedule growth: An update,” in *2010 IEEE Aerospace Conference*, pp. 1–12.
- [17] BLANCHARD, B. and FABRYCKY, W., *Systems Engineering and Analysis*. Prentice Hall, 1998.
- [18] BOZE, W. and HEANEY, E., “An expanded study of sawe paper 3468 - quantifying uncertainty and risk in vehicle mass properties throughout the design development phase,” *Weight Engineering*, vol. 70, no. 2, 2010.
- [19] BOZE, W. and HESTER, P., “Quantifying uncertainty and risk in vehicle mass properties throughout the design development phase,” *Weight Engineering*, vol. 69, no. 1, 2009. This talks about a breakdown by uncertainty classification and introduces the Min, most likely, and max treatments for mass allocations.
- [20] BRADFORD, J. and HELLMAN, B., “Return to Launch Site Trajectory Options for a Reusable Booster without a Secondary Propulsion System,” in *AIAA Space 2009 Conference and Exposition*, 2009.
- [21] BURROUGHS, S. and JUNTIMA, G., “Exploring techniques for contingency setting,” in *AACE International Transactions*.
- [22] CARPENTER, J., “Fast design and operability experiment.” March 2008.
- [23] CENTER, M. S. F., “Weight control and weight histories of marshall space flight center projects,” tech. rep., National Aeronautics and Space Administration, 1991.
- [24] CHAI, P. and WILHITE, A., “Quantifying the Effects of Model Uncertainty on Design Mass Margin in Advanced Earth-to-Orbit Launch Vehicles,” in *AIAA Space 2010 Conference and Exposition*, 2010.
- [25] CHAPMAN, C., WARD, S., and WARD, S., *Project risk management: processes, techniques, and insights*. John Wiley and Sons Inc, 2003.
- [26] CHAU, K., “Monte carlo simulation of construction costs using subjective data,” *Construction Management and Economics*, vol. 13, no. 5, pp. 369–383, 1995.
- [27] CHUNG, L. and DO PRADO LEITE, J., “On non-functional requirements in software engineering,” *Conceptual modeling: Foundations and applications*, pp. 363–379, 2009.
- [28] CURRAN, M., “Range estimating measuring uncertainty and reasoning with risk,” *Cost Engineering*, vol. 31, no. 3, pp. 18–26, 1989.
- [29] CURTIS, H., *Orbital Mechanics for Engineering Students*. 2005.
- [30] DE ROCQUIGNY, E., DEVICTOR, N., and TARANTOLA, S., *Uncertainty in industrial practice*. John Wiley and Sons, Ltd, 2008.
- [31] DELAURENTIS, D., *A Probabilistic Approach to Aircraft Design Emphasizing Stability and Control Uncertainties*. PhD thesis, Georgia Institute of Technology, 1998.
- [32] DEPARTMENT OF DEFENSE, “Mass properties control for space vehicles,” 1998.

- [33] DEPARTMENT OF DEFENSE, *Manufacturing Readiness Level Deskbook Version 2.0*. 2011.
- [34] DIEKMANN, J., “Probabilistic estimating: mathematics and applications,” *Journal of Construction Engineering and Management*, vol. 109, no. 3, pp. 297–308, 1983.
- [35] DISSEL, A. F., HUSEMAN, P. G., BOMBA, J. V., MAYFIELD, D. P., and HELLMAN, B. M., “Aft end heating assessment of multiple engine clusters for a reusable first stage launch vehicle,” in *AIAA Space 2012 Conference & Exposition*, 2012.
- [36] DISSEL, A., KOTHARI, A., LIVINGSTON, J., and LEWIS, M., “Weight growth study of reusable launch vehicle systems,” *Journal of Spacecraft and Rockets*, vol. 44, no. 3, pp. 640–648, 2007.
- [37] DUBOS, G., SALEH, J., and BRAUN, R., “Technology readiness level, schedule risk, and slippage in spacecraft design,” *Journal of Spacecraft and Rockets*, vol. 45, pp. 836–842, 2008.
- [38] ENCYCLOPEDIA ASTRONAUTICA, “Atlas v,” 2011.
- [39] ENCYCLOPEDIA ASTRONAUTICA, “Delta iv,” 2011.
- [40] ENGLER III, W., BILTGEN, P., and MAVRIS, D., “Concept Selection Using an Interactive Reconfigurable Matrix of Alternatives (IRMA),” in *45th AIAA Aerospace Sciences Meeting and Exhibit*, 2007.
- [41] FAHEY, L. and RANDALL, R., *Learning from the future: competitive foresight scenarios*. John Wiley and Sons Inc, 1998.
- [42] FISCHER, G., “When oracles fail—a comparison of four procedures for aggregating subjective probability forecasts,” *Organizational Behavior and Human Performance*, vol. 28, no. 1, pp. 96–110, 1981.
- [43] FLYVBJERG, B., “From nobel prize to project management: Getting risks right,” *Project Management Journal*, vol. 37, no. 3, p. 5, 2006.
- [44] FREANER, C., BITTEN, R., BEARDEN, D., and EMMONS, D., “An assessment of the inherent optimism in early conceptual designs and its effect on cost and schedule growth, presentation to the planetary science subcommittee, nasa advisory council,” 2008.
- [45] GAMMA, E., HELM, R., JOHNSON, R., and VLISSIDES, J., *Design patterns: Elements of reusable object-oriented design*. Addison-Wesley Reading, 1995.
- [46] GARVEY, P., *Probability Methods for Cost Uncertainty Analysis: A Systems Engineering Perspective*. New York: Marcel Dekker Inc., 2000.
- [47] GRADY, J., *System Requirements Analysis*. McGraw-Hill, Inc., 1993.
- [48] GRANTZ, A., “X-37B Orbital Test Vehicle and Derivatives,” in *AIAA Space 2011 Conference and Exposition*, 2011.
- [49] GRIFFIN, M. and FRENCH, J., *Space Vehicle Design*. Reston, VA: American Institute of Aeronautics and Astronautics, Inc., 2004.

- [50] GUO, J. and DU, X., “Sensitivity analysis with mixture of epistemic and aleatory uncertainties,” *AIAA Journal*, vol. 45, no. 9, 2007.
- [51] HAMPSTEN, K. and HICKMAN, R., “Next Generation Air Force Spacelift,” in *AIAA Space 2010 Conference and Exposition*, 2010.
- [52] HAMPTON, J., MOORE, P., and THOMAS, H., “Subjective probability and its measurement,” *Journal of the Royal Statistical Society. Series A (General)*, pp. 21–42, 1973.
- [53] HAWKINS, K., “Space Vehicle and Associated Subsystem Weight Growth,” in *47th Annual Conference of the Society of Allied Weight Engineers, Inc.*, 1988.
- [54] HEINEMAN, W., “Design mass properties ii mass estimating and forecasting for aerospace vehicles based on historical data,” tech. rep., National Aeronautics and Space Administration, Lyndon B. Johnson Space Center, 1994.
- [55] HELLMAN, B., NGO, A., and WALLACE, J., “Technical Challenges for an Integrated Reusable Booster System Flight Demonstrator ,” in *AIAA Space 2010 Conference and Exposition*, 2010.
- [56] HEPPEHEIMER, T., *History of the Space Shuttle: Volume Two, Development of the Space Shuttle, 1972–1981*. Washington, DC: Smithsonian Institution Press, 2002.
- [57] HILL, P. and PETERSON, C., *Mechanics and Thermodynamics of Propulsion*. Addison-Wesley, 1992.
- [58] HINES, W. W., MONTGOMERY, D. C., and BORROR, D. M. G. C. M., *Probability and statistics in engineering*. John Wiley & Sons, 2008.
- [59] HOLLMANN, J., “Recommended Practices for Risk Analysis and Cost Contingency Estimating,” in *2009 ACE International Transactions*, 2009.
- [60] HOLLMANN, J., “The monte-carlo challenge: A better approach,” *Proceedings of 2007 ACE International Transactions*, pp. 03.01–03.07, 2007.
- [61] JENKINS, D., *Space Shuttle: The History of the National Space Transportation System The First 100 Missions, 3rd Edition*. 2001.
- [62] JENKINS, D., “Broken in midstride space shuttle as a launch vehicle,” in *To Reach the High Frontier* (LAUNIUS, R. and JENKINS, D., eds.), The University Press of Kentucky, 2002.
- [63] JIMENEZ, H. and MAVRIS, D., “An Evolution of Morphological Analysis Applications in Systems Engineering,” in *48th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2010.
- [64] JIMENEZ, H., STULTS, I., and MAVRIS, D., “A Morphological Approach for Proactive Risk Management in Civil Aviation Security,” in *47th AIAA Aerospace Sciences Meeting Including the New Horizons Forum and Aerospace Exposition*, 2009.
- [65] KAHN, H. and WIENER, A., “The next thirty-three years: a framework for speculation,” *Daedalus*, vol. 96, no. 3, pp. 705–732, 1967.

- [66] KAHN, H., WIENER, A., and INSTITUTE, H., *The year 2000: a framework for speculation on the next thirty-three years*, vol. 2. Macmillan New York, 1967.
- [67] KAHNEMAN, D. and TVERSKY, A., “Intuitive prediction: Biases and corrective procedures,” in *Studies in the management sciences: Forecasting* (MAKRIDAKIS, S. and WHEELWRIGHT, S., eds.), 1979.
- [68] KAHNEMAN, D. and TVERSKY, A., “Prospect theory: An analysis of decision under risk,” *Econometrica: Journal of the Econometric Society*, pp. 263–291, 1979.
- [69] KAYTON, M. and FRIED, W., *Avionics Navigation Systems*. John Wiley and Sons, Inc., 1997.
- [70] KENNEDY SPACE CENTER, “National space transportation system (nsts) shuttle reference manual,” tech. rep., NASA, 1988.
- [71] KETSDEVER, A., YOUNG, M., MOSSMAN, J., and PANCOTTI, A., “Overview of advanced concepts for space access,” *Journal of Spacecraft and Rockets*, vol. 47, no. 2, pp. 238–250, 2010.
- [72] KRZYSZTOFOWICZ, R., “The case for probabilistic forecasting in hydrology,” *Journal of Hydrology*, vol. 249, no. 1, pp. 2–9, 2001.
- [73] KUPRENAS, J. and NASR, E., “Controlling design-phase scope creep,” *AACE International Transactions*, 2003.
- [74] LARSON, W. and PRANKE, L., *Human Spaceflight: Mission Analysis and Design*. McGraw-Hill, 2000.
- [75] LAW, A., *Simulation modeling and analysis*. Boston, MA [etc.]: McGraw-Hill, 2007.
- [76] LETCHWORTH, G., “X-33 Reusable Launch Vehicle Demonstrator, Spaceport and Range,” in *AIAA SPACE 2011 Conference and Exposition*, 2011.
- [77] LIVINGSTON, J., “Comparative Analysis of Rocket and Air-breathing Launch Vehicles,” in *Space 2004 Conference and Exhibit*, 2004.
- [78] LOVALLO, D. and KAHNEMAN, D., “Delusions of success,” *Harvard business review*, vol. 81, no. 7, pp. 56–63, 2003.
- [79] MAACK, J., “5. scenario analysis: A tool for task managers,” *Social Analysis Selected Tools and Techniques*, p. 62, 2001.
- [80] MAK, S. and PICKEN, D., “Using risk analysis to determine construction project contingencies,” *Journal of Construction Engineering and Management*, vol. 126, p. 130, 2000.
- [81] MANKINS, J., “Technology readiness levels,” *White Paper, April*, vol. 6, 1995.
- [82] MATHEWS, G., “Weight growth: A process and its utility,” in *53rd Annual Conference of the Society of Allied Weight Engineers, Inc.*, 1994.
- [83] MECHAM, M., ANSELMO, J., and NORRIS, G., “A wing and a prayer,” *Aviation Week and Space Technology*, vol. 171, no. 4, pp. 22–24, 2009.

- [84] MELCHERS, R., *Structural Reliability Analysis and Prediction, 2nd Ed.* John Wiley and Sons, 1999.
- [85] MERROW, E., CHAPEL, S., and WORTHING, C., “Review of cost estimation in new technologies: implications for energy process plants,” tech. rep., RAND Corp., Santa Monica, CA (USA), 1979.
- [86] MEYER, M. A. and BOOKER, J. M., *Eliciting and analyzing expert judgment: a practical guide*, vol. 7. SIAM, 1991.
- [87] MONEY, S., “Competition and the future of the eelv program,” *The Space Review*, 2011.
- [88] MONROE, R. W., LEPSCH JR, R. A., and UNAL, R., “Using expert judgment methodology to address uncertainty in launch vehicle weight estimates,” in *9th AIAA/ISSMO Symposium and Exhibit on Multidisciplinary Analysis and Optimization, Atlanta, GA*, 2002.
- [89] MUIRHEAD, B., SHISHKO, R., and FOX, G., “Stochastic analysis of constellation performance and mass margins,” *Journal of Spacecraft and Rockets*, vol. 46, no. 5, pp. 1077–1088, 2009.
- [90] MYERS, R. H. and MONTGOMERY, D. C., *Response surface methodology*. Taylor & Francis, 1988.
- [91] NATIONAL AERONAUTICS AND SPACE ADMINISTRATION, “Shuttle systems weight and performance monthly status report december 15, 1982,” tech. rep., 1982.
- [92] NATIONAL AERONAUTICS AND SPACE ADMINISTRATION, “Nasa systems engineering handbook,” Tech. Rep. NASA/SP-2007-6105 Rev 1, National Aeronautics and Space Administration, 2007.
- [93] NATIONAL AERONAUTICS AND SPACE ADMINISTRATION, GLENN RESEARCH CENTER, “Rules for the design, development, verification, and operation of flight systems,” tech. rep., 2005.
- [94] NORRIS, K. W., “The morphological approach to engineering design,” in *Conference on Design Methods* (JONES, J. and THORNLEY, D., eds.), pp. 115–140, Pergamon Press, Incorporated.
- [95] NORTH AMERICAN ROCKWELL SPACE DIVISION, “Space shuttle mass properties status report, july 1973,” Tech. Rep. N76-79009, 1973.
- [96] NORTH AMERICAN ROCKWELL SPACE DIVISION, “Space shuttle mass properties status report, may 1973,” Tech. Rep. N76-79428, 1973.
- [97] NORTH AMERICAN ROCKWELL SPACE DIVISION, “Space shuttle mass properties status report, february 1974,” Tech. Rep. N77-70413, 1974.
- [98] NORTHROP GRUMMAN CORPORATION, “Fast design and operability experiments—vehicle maturation.” 2008.

- [99] OBERKAMPF, W., DELAND, S., RUTHERFORD, B., DIEGERT, K., and ALVIN, K., “Error and uncertainty in modeling and simulation,” *Reliability Engineering and System Safety*, vol. 75, no. 3, pp. 333–357, 2002.
- [100] PAYAN, A., *Enabling methods for the design and optimization of detection architectures*. PhD thesis, Georgia Institute of Technology, 2013.
- [101] PESSIN, M., “Lessons learned from space shuttle external tank development- a technical history of the external tank,” tech. rep., Marshall Space Flight Center, 2002.
- [102] PETERSON, J., RATZ, O. G., and TREGO, A., “Development and Implementation of a Space Systems Mass Properties Process,” in *67th Annual Conference of Society of Allied Weight Engineers, Inc.*, 2008.
- [103] PILET, J., DIECIDUE-CONNERS, D., WORDEN, M., GUILLOT, M., and WELZYN, K., “External Tank Program – Legacy of Success,” in *AIAA Space 2011 Conference and Exposition*, 2011.
- [104] PRESSMAN, R. S., *Software engineering: a practitioner’s approach*, vol. 5. McGraw-hill New York, 1992.
- [105] RAND, “National security space launch report,” 2006.
- [106] RASKY, D., KOLODZIEJ, P., FARKAS, S., and DUTRO, L., “The phased development approach: for advanced technology and complex systems,” in *41st Aerospace Sciences Meeting and Exhibit, Reno, Nevada*, 2003.
- [107] RAYMER, D., *Aircraft design: a conceptual approach*. Reston, VA: American Institute of Aeronautics and Astronautics, 1999.
- [108] REGAN, H., COLYVAN, M., and BURGMAN, M., “A taxonomy and treatment of uncertainty for ecology and conservation biology,” *Ecological applications*, vol. 12, no. 2, pp. 618–628, 2002.
- [109] RIERA, J., ROCHA, M., and RAMOS DE MENEZES, R., “On the consideration of phenomenological uncertainty,” *Nuclear engineering and design*, vol. 175, no. 3, pp. 259–265, 1997.
- [110] RITCHEY, T., “Problem structuring using computer-aided morphological analysis,” *Journal of the Operational Research Society*, vol. 57, no. 7, pp. 792–801, 2006.
- [111] RITCHEY, T., *Wicked Problems - Social Messes: Decision Support Modelling with Morphological Analysis*. Berlin, Heidelberg: Springer Verlag, 2011.
- [112] ROBERTSON, B., CROWLEY, D., DOUGLAS, R., MAVRIS, D., and HELLMAN, B., “Configuration determination of a reusable first stage booster utilizing monte carlo based inverse design,” in *AIAA Space 2011 Conference & Exposition*, 2011.
- [113] ROBERTSON, B., CROWLEY, D., GARMENDIA, D., DOUGLAS, R., MAVRIS, D. N., and HELLMAN, B., “A modeling methodology and environment for integrated trim analysis for a reusable booster system,” in *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition. Orlando, FL*, pp. 2011–1009, 2011.

- [114] ROHRSCHEIDER, R. and OLDS, J., "A Comparison of Modern and Historic Mass Estimating Relationships on a Two Stage to Orbit Launch Vehicle," in *AIAA Space 2001 Conference and Exposition*, 2001.
- [115] RYAN, R., "A history of aerospace problems, their solutions, their lessons," tech. rep., National Aeronautics and Space Administration, 1996.
- [116] RYAN, R. and TOWNSEND, J., "Fundamentals and issues in launch vehicle design," *Journal of Spacecraft and Rockets*, vol. 34, pp. 192–198, 1997.
- [117] SAS INSTITUTE INC, *JMP 10 Modeling and Multivariate Methods*. SAS Institute Inc, 2012.
- [118] SAUVAGEAU, D. and ALLEN, B., "Launch Vehicle Historical Reliability," in *34th Joint Propulsion Conference and Exhibit*, 1998.
- [119] SCHNAARS, S., "How to develop and use scenarios," *Long range planning*, vol. 20, no. 1, pp. 105–114, 1987.
- [120] SCHREINER, W., HUNT, D., ROCKEN, C., and SOKOLOVSKIY, S., "Precise GPS Data Processing for the GPS/MET Radio Occultation Mission at UCAR," in *National Technical Meeting of The Institute of Navigation*, pp. 103–112, 1998.
- [121] SCHULTZ, T., "The Use of Contingencies in Predicting and Controlling Vehicle Weight Growth from a Missile Perspective," in *52nd Annual Conference of the Society of Allied Weight Engineers, Inc.*, 1993.
- [122] SCHUSTER, A., "Weight Risk Using Uncertainty Analysis for a Marine System," in *62nd Annual Conference of Society of Allied Weight Engineers, Inc.*, 2003.
- [123] SCHUSTER, A., "Weight Risk Using Monte Carlo Analysis for a Marine System," in *64th Annual Conference of the Society of Allied Weight Engineers, Inc.*, 2005.
- [124] SCHUSTER, A., "A Method to Develop a Margin Plan," in *66th Annual Conference of the Society of Allied Weight Engineers, Inc.*, 2007.
- [125] SOCIETY OF ALLIED WEIGHT ENGINEERS INC., "Mass properties control for space vehicles," 2000.
- [126] TELLET, D., "Submarine Margin Determination by Stochastic Simulation," in *Regional Conference of the Society of Allied Weight Engineers, Inc.*, 2006.
- [127] TEWARI, A., *Atmospheric and space flight dynamics: modeling and simulation with MATLAB and Simulink*. Springer, 2007.
- [128] THAL JR, A., COOK, J., and WHITE III, E., "Estimation of cost contingency for air force construction projects," *Journal of Construction Engineering and Management*, vol. 136, p. 1181, 2010.
- [129] THOMPSON, P. and PERRY, J., *Engineering construction risks: A guide to project risk analysis and assessment implications for project clients and project managers*. Thomas Telford, 1992.

- [130] THOMPSON, R., WILHITE, A., REEVES, D., STANLEY, D., and WAGNER, J., “Mass growth in space vehicle and exploration architecture development,” *Acta Astronautica*, vol. 66, no. 7-8, pp. 1220–1236, 2010.
- [131] THUNNISSEN, D., “Method for determining margins in conceptual design,” *Journal of Spacecraft and Rockets*, vol. 41, no. 1, pp. 85–92, 2004.
- [132] THUNNISSEN, D., *Propagating and Mitigating Uncertainty in the Design of Complex Multidisciplinary Systems*. PhD thesis, California Institute of Technology, 2005.
- [133] TURNER, A., BALESTRINI-ROBINSON, S., and MAVRIS, D., “Heuristics for the regression of stochastic simulations,” *Journal of Simulation*, 2013.
- [134] UNITED LAUNCH ALLIANCE, “Atlas v user’s guide,” 2010.
- [135] UNITED STATES AIR FORCE, “Baa 06-01 pkd, reusable operoperation responsive trans-asmospheric technology call 3, fully reusable acces to space technology (fast) ground experiments,” 2007.
- [136] UNITED STATES GOVERNMENT ACCOUNTABILITY OFFICE, “Defense acquisitions: Assessments of selected weapon programs,” Tech. Rep. GAO-09-326SP, 2009.
- [137] UNITED STATES GOVERNMENT ACCOUNTABILITY OFFICE, “Defense acquisitions: Assessments of selected weapon programs,” Tech. Rep. GAO-10-388SP, 2010.
- [138] UNITED STATES GOVERNMENT ACCOUNTABILITY OFFICE, “Defense acquisitions: Assessments of selected weapon programs,” Tech. Rep. GAO-11-233SP, 2011.
- [139] VOSE, D., *Risk analysis: a quantitative guide*. John Wiley and Sons Inc, 2008.
- [140] WALLACE, J., ST. GERMAIN, B., and CHARANIA, A., “Probabilistic Examination of Lunar Landers,” in *Space 2005*, 2005.
- [141] WALPOLE, R., MYERS, R., MYERS, S., and YE, K., *Probability and statistics for engineers and scientists*. Upper Saddle River, NJ: Prentice Hall, 8 ed., 1998.
- [142] WALTON, M., *Managing uncertainty in space systems conceptual design using portfolio theory*. PhD thesis, 2002.
- [143] WERTZ, J. and LARSON, W., *Space Mission Analysis and Design*. Microcosm and Kluwer, 1999.
- [144] WHITE, T. A., “RS-68 Weight History and Predicted Flight Weight,” in *58th Annual Conference of the Society of Allied Weight Engineers, Inc.*, 1999.
- [145] WHITMORE, S. and DUNBAR, B., “Orbital Space Plane: Past Present, and Future,” in *AIAA/ICAS International Air and Space Symposium and Exposition: The Next 100 Years*, 2003.
- [146] WILHITE, A., GHOLTSON, S. E., FARRINGTON, P. A., and SWAIN, J. J., “Evaluating technology projections and weight prediction method uncertainty of future launch vehicles,” *Journal of Spacecraft and Rockets*, vol. 45, no. 3, pp. 587–591, 2008.

- [147] WILLIAMS, G., “Weight risk analysis,” in *67th Annual Conference of Society of Allied Weight Engineers, Inc.*, 1978.
- [148] WILLIAMSON, R. A., “The biggest of them all reconsidering the saturn v,” in *To Reach the High Frontier* (LAUNIUS, R. and JENKINS, D., eds.), The University Press of Kentucky, 2002.
- [149] WILLIAMSON, R. A. and LAUNIUS, R., “Rocketry and the origins of space flight,” in *To Reach the High Frontier* (LAUNIUS, R. and JENKINS, D., eds.), The University Press of Kentucky, 2002.
- [150] YARNELL, N., “Design, verification/validation and operations principles for flight systems,” 2003.
- [151] YEO, K., “Risks, classification of estimates, and contingency management,” *Journal of Management in Engineering*, vol. 6, p. 458, 1990.
- [152] ZIO, E. and APOSTOLAKIS, G., “Two methods for the structured assessment of model uncertainty by experts in performance assessments of radioactive waste repositories,” *Reliability Engineering and System Safety*, vol. 54, no. 2, pp. 225–241, 1996.
- [153] ZWICKY, F., “Morphology and nomenclature of jet engines,” *Aeronautical Engineering Review*, vol. 6, no. 6, pp. 49–50, 1947.

VITA

Bradford Elliott Robertson was born on June 12th, 1983 in Metairie, LA to Wayne and Laura Robertson. He attended Jesuit High School in New Orleans, LA before enrolling in the Computer Science program at the Georgia Institute of Technology in 2001. As an undergraduate, Mr. Robertson worked as an undergraduate research assistant under Dr. Dimitri Mavris at the Aerospace Systems Design Laboratory (ASDL). As an undergraduate research assistant, he became interested in aerospace problems and enrolled in the Aerospace Engineering program at Georgia Tech. He graduated with highest honors in 2007 with a B.S. in both Computer Science and Aerospace Engineering. After accepting a position as a graduate researcher at ASDL, he graduated with a M.S. in Aerospace Engineering in 2009. Over the past ten years Mr. Robertson has worked with a number of industry and government sponsors in the areas of missile design, missile defense, manned spacecraft, and reusable launch vehicles.