CONTRAST: A CONCEPTUAL RELIABILITY GROWTH APPROACH FOR COMPARISON OF LAUNCH VEHICLE ARCHITECTURES

A Thesis
Presented to
The Academic Faculty

by

Mathew R. Zwack

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

Georgia Institute of Technology
December 2014

CONTRAST: A CONCEPTUAL RELIABILITY GROWTH APPROACH FOR COMPARISON OF LAUNCH VEHICLE ARCHITECTURES

Approved by:

Professor Dimitri Mavris, Advisor
School of Aerospace Engineering
Georgia Institute of Technology

Professor Daniel Schrage
School of Aerospace Engineering
Georgia Institute of Technology

Mr. Reginald Alexander
Marshall Space Flight Center
National Aeronautics and Space Administration

Dr. Vitali Volovoi

Dr. Jean Charles Domercant
School of Aerospace Engineering
Georgia Institute of Technology

Date Approved: November 17, 2014

To my wife,

Kathryn K. Zwack

ACKNOWLEDGEMENTS

I would like to take this opportunity to thank all those who have helped me through this process. First, a very large thank you is due to my committee members; Dr. Dimitri Mavris, Dr. Daniel Schrage, Mr. Reginald Alexander, Dr. Vitali Volovoi, and Dr. Jean Charles Domercant. I appreciate the all of the time you have taken to help guide me through the process of completing my thesis. All of your comments and questions have been instrumental in shaping this document into a final product that I am proud to present. I would like to extend a special thanks to my advisor, Dr. Mavris, for allowing me to join such a great team here at ASDL. Your dedication to the success of this lab has allowed me the opportunity to work on a wide variety of interesting research projects. Outside of the academics and research, your meetings have taught me so much about what it means to earn and hold a Ph.D. Thank you Dr. Mavris; your input has shaped me in many ways, which will be of great help as I move on from ASDL.

In addition to my committee I would like to thank my research engineer, Stephen Edwards, for helping to shape this research during its earliest phases. I appreciate your willingness to sit down with me to discuss my topic, even though it may have delayed the progress on your own. Your flexibility and understanding in regard to our sponsored research work has also helped me complete this thesis within my very aggressive timeline.

I also owe thanks to my colleagues in ASDL for their input and feedback over the past few years. First I would like to thank my classmates, whose camaraderie has helped me through all of the hurdles of this process; from studying for the qualifying exams to the preparation for my proposal and defense. To the Fall 2012 quals group, Thank You! Next I would like to thank all of those who sat with me in the basement ITAR bay, both past and present. Thank you to the older students; Dr. Jonathan Sharma, Dr. Bradford Robertson, Dr. Andrew Turner, Nick Molino, and Dan Garmendia for being available to field general questions and give feedback on my presentations and document; and thank you to the others, Michael Steffens, Blaine Laughlin, Scott Strong, John Dykes, and John Robinson who provided feedback, were instrumental in keeping me motivated, and supplied extra conversation and comic relief to get through those days where nothing seemed to work.

Next I would like to thank my family for their support through my years as a graduate student. I owe my deepest gratitude to my wife Kelsey, who has put up with my long hours of work while finishing up this document. I appreciate your daily encouragement and your willingness to take care of things outside of work so I could focus on writing. I would also like to thank my parents Paul and Carol Zwack for their love and support. While growing up you helped instill the work ethic that I have today. Through your words of encouragement you kept me motivated and on track to reach my goals.

Finally, I would like to extend thanks to those outside of ASDL who have helped me along the way. First, I want to thank the Western Golf Association Evans Scholars Foundation for providing me with an undergraduate education that was a huge step towards my graduation from Georgia Tech. Without the support from the Evans program, its donors, and board members this would not have been possible. Thanks also to my good friends and classmates from University of Minnesota including Cole Kazemba, Kyle Zakrzewski, Cory Gloe, and Phil Courey. Working and studying with you solidified the work habits that have been invaluable to me throughout the entire Ph.D. process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACERT	Advanced Concepts Evaluating Risk Tool
AMSAA	Army Materiel Systems Analysis Activity
ASDL	Aerospace Systems Design Laboratory
ATK	Alliant Tech Systems
CCF	Common Cause Failure
CER	Concept Exploration and Refinement
CIL	Critical Items List
CONTRAST	Conceptual Reliability Growth Approach for Comparison of Launch Vehicle Architectures
CRM	Continuous Risk Management
DoD	United States Department of Defense
ECSS	European Cooperation for Space Standardization
ET	Space Shuttle External Tank
FEF	Fix Effectiveness Factor
FIRST	Flight-oriented Integrated Reliability and Safety Tool
FMEA	Failure Mode and Effect Analysis
FMECA	Failure Mode, Effect, and Criticality Analysis
FTA	Fault Tree Analysis
GNC	Guidance, Navigation, and Control
GUI	Graphical User Interface
IOA	Independent Orbiter Assessment
IPPD	Integrated Product and Process Development
IRMA	Interactive Reconfigurable Matrix of Alternatives
IVF	Integrated Vehicle Fluids
LH2	Liquid Hydrogen

LOC	Loss of Crew
LOM	Loss of Mission
LOV	Loss of Vehicle
LOX	Liquid Oxygen
LRB	Liquid Rocket Booster
LV	Launch Vehicle
MC	Monte Carlo Simulation
MOA	Matrix of Alternatives
MPS	Main Propulsion System
MSFC	Marshall Space Flight Center
NASA	National Aeronautics and Space Administration
PCM	Parts Count Method
PDR	Preliminary Design Review
PHA	Preliminary Hazard Analysis
PRA	Probabilistic Risk Assessment
RBD	Reliability Block Diagram
RIDM	Risk Informed Decision Making
SAIC	Science Applications International Corporation
SLS	Space Launch System
SME	Subject matter expert
SPN	Stochastic Petri Net
SRB	Solid Rocket Booster
SSME	Space Shuttle Main Engine
STS	Space Transportation System
ULA	United Launch Alliance
USAF	United States Air Force

SUMMARY

In 2004, the NASA Astronaut Office produced a memo regarding the safety of next generation launch vehicles. The memo requested that these vehicles have a probability of loss of crew of at most 1 in 1000 flights, which represents nearly an order of magnitude decrease from current vehicles. The goal of LOC of 1 in 1000 flights has since been adopted by the launch vehicle design community as a requirement for the safety of future vehicles. This research addresses the gap between current vehicles and future goals by improving the capture of vehicle architecture effects on reliability and safety.

Vehicle architecture pertains to the physical description of the vehicle itself, which includes manned or unmanned, number of stages, number of engines per stage, engine cycle types, redundancy, etc. During the operations phase of the vehicle life-cycle it is clear that each of these parameters will have an inherent effect on the reliability and safety of the vehicle. However, the vehicle architecture is typically determined during the early conceptual design phase when a baseline vehicle is selected. Unless a great amount of money and effort is spent, the architecture will remain relatively constant from conceptual design through operations. Due to the fact that the vehicle architecture is essentially "locked-in" during early design, it is expected that much of the vehicle's reliability potential will also be locked-in.

This observation leads to the conclusion that improvement of vehicle reliability and safety in the area of vehicle architecture must be completed during early design. Evaluation of the effects of different architecture decisions must be performed prior to baseline selection, which helps to identify a vehicle that is most likely to meet the reliability and safety requirements when it reaches operations. Although methods

exist for evaluating reliability and safety during early design, weaknesses exist when trying to evaluate all architecture effects simultaneously.

The goal of this research was therefore to formulate and implement a method that is capable of quantitatively evaluating vehicle architecture effects on reliability and safety during early conceptual design. The ConcepTual Reliability Growth Approach for CompariSon of Launch Vehicle ArchiTectures (CONTRAST) was developed to meet this goal. Using the strengths of existing techniques a hybrid approach was developed, which utilizes a reliability growth projection to evaluate the vehicles. The growth models are first applied at the subsystem level and then a vehicle level projection is generated using a simple system level fault tree. This approach allows for the capture of all trades of interest at the subsystem level as well as many possible trades at the assembly level.

The CONTRAST method is first tested on an example problem, which compares the method output to actual data from the Space Transportation System (STS). This example problem illustrates the ability of the CONTRAST method to capture reliability growth trends seen during vehicle operations. It also serves as a validation for the development of the reliability growth model assumptions for future applications of the method.

The final chapter of the thesis applies the CONTRAST method to a relevant launch vehicle, the Space Launch System (SLS), which is currently under development. Within the application problem, the output of the method is first used to check that the primary research objective has been met. Next, the output is compared to a state-of-the-art tool in order to demonstrate the ability of the CONTRAST method to alleviate one of the primary consequences of using existing techniques. The final section within this chapter presents an analysis of the booster and upper stage block upgrade options for the SLS vehicle. A study of the upgrade options was carried out because the CONTRAST method is uniquely suited to look at the effects of such strategies. The results from the study of SLS block upgrades give interesting observations regarding the desired development order and upgrade strategy. Ultimately this application problem demonstrates the merits of applying the CONTRAST method during early design. This approach provides the designer with more information in regard to the expected reliability of the vehicle, which will ultimately enable the selection of a vehicle baseline that is most likely to meet the future requirements.

CHAPTER I

INTRODUCTION

1.1 Motivation

Launch vehicles are complex system of systems used to transport expensive, often one-of-a-kind payloads to earth orbit and beyond. These vehicles rely on a multitude of different subsystems spanning a wide variety of engineering disciplines, such as: aerodynamics, propulsion, structures, controls, etc. Due to the extreme nature of the environment in which these systems operate, they are inherently sensitive to outside perturbations, component defects, and faults. Even the slightest defect or fault in one of the cheapest components has the potential to cause a loss of mission or loss of vehicle. Therefore, the reliability and safety of launch vehicles is of utmost importance due to large impacts on cost, schedule, and risk.

First, the design, development, and operations costs associated with new launch vehicles are immense. For example, the SpaceX Falcon 9, which is touted as a "cheaper" alternative, has an estimated development cost of between \$1B-\$3B [80, 118]. Another example of this high cost is the price paid to launch a given payload, which ranges from \$50M-\$500M per launch or \$20k-\$30k per kilogram [28, 139].

Along with the high price paid to launch a payload, it is important to note the often one-of-a-kind nature of these payloads. This is especially true for manned spaceflight, where loss of crew is a very large consideration. For non-manned flights, large satellites (communications, surveillance, scientific, etc.) or interplanetary missions can easily cost in the hundreds of millions to billions of dollars and take up to 10 years to produce [19, 26, 146]. With so much at stake, the risk of losing such a payload is an excellent example of why launch vehicles must be highly reliable.

The un-reliability of a launch vehicle can also directly affect cost through changes in schedule. A launch vehicle failure can cause delays in future launches, requiring costly investigations and possibly major redesigns [166]. Failures can also affect the launch schedule in terms of cancellation of future flights. This can be due to either loss of customers (commercial LVs) or complete cancellation of the program (government/military LVs) [29, 82]. An excellent example of the effects of LV failures is the company Sea Launch, which lost a $400M satellite in 2013 after recovering from filing for bankruptcy in 2009 [29, 39].

Over the past 60 years, the United States has had a variety of different launch vehicles spanning from the many failures of the Vanguard program to the great success of current vehicles such as the Delta or Atlas [30, 63, 141]. Historically, it has been estimated that the combined overall probability of success for U.S. launch vehicles is between 85 and 90% [25, 93]. It is interesting to think about the reliability of launch vehicles in this sense, as a 1 in 10 chance of LOM is quite high considering the value of a typical payload. For manned launch vehicles these stakes are even higher, requiring vehicles with the highest reliability and safety.

The Russian Soyuz, which is currently used to ferry Astronauts and Cosmonauts to and from the International Space Station, is considered to be the most reliable vehicle in operation today [87]. The estimated probability of LOC for Soyuz is 1 in 400 flights [59]. The most recent American manned launch vehicle, STS (also known as the Space Shuttle), had an estimated probability of LOC of 1 in 100 flights at the end of its lifetime [61]. On ascent the Space Shuttle program only encountered one catastrophic failure with the loss of Challenger in 1986 [94]. Thus, its demonstrated safety was actually a bit higher than 1 in 100.

Although the safety numbers for recent manned launch vehicles are respectable, future goals aim to increase vehicle safety by an order of magnitude. In 2004, the NASA Astronaut Office produced a memo regarding the risk to crews on future launch vehicles. This memo requested that the probability of LOC be at most 1 in 1000 flights for next generation vehicles [105]. Since the release of the memo, many authors have acknowledged 1 in 1000 as a requirement to be used in the assessment of new vehicles [16, 59, 61, 142, 153].

The new goal of 1 in 1000 flights brings to light a large gap between current technology and future requirements. From this gap, two motivating questions were derived to guide the research. The first asks: What drives reliability and safety throughout a launch vehicle's life-cycle? The second pertains to the gap itself, asking: Where can this increase in reliability and safety come from? In order to answer these questions, a list of drivers for reliability and safety was developed from the literature, which can be divided into three general categories: operating environment, programmatic environment, and vehicle architecture.

The first category, operating environment, includes operational considerations leading up to a launch as well as noise during vehicle operations. Factors such as component storage, vehicle shipping, and vehicle assembly are often the topics of launch site operations research [22, 38, 68, 69]. During transportation, events such as shock and vibration, as well as thermal transients must be taken into account [63, 94]. Such events can have a negative effect on vehicle components, and may cause early wear out or failure. Storage effects such as shelf life must also be taken into account for components that will not be integrated into the vehicle for an extended period of time [63, 94].

After the components have arrived on site, the proper assembly of the vehicle is of utmost importance. Errors may occur during assembly that will cause failure of the vehicle during launch [63, 129]. A recent example of the consequences of miss-assembly is the failure of a Proton launch vehicle on July 2, 2013. In this case a set of three yaw angular rate sensors were installed upside down, which caused the failure of the vehicle upon launch [157].

Another operating environment consideration is the launch trajectory. On the day of launch, the vehicle's trajectory is designed in order to ensure that its structural capability is not exceeded [13]. The major concern with trajectory design is the launch site winds and winds aloft [139]. The scrubbing of a launch due to winds beyond allowable levels is a very common occurrence in the world of launch vehicles [13, 139].

Improvements to vehicle reliability in the area of operating environment can be made primarily by adhering to strict operational policy. For example, to avoid failures caused by shipping effects or assembly errors vehicle inspection is critical [24, 63, 91, 94, 129]. Inspections can help to identify potential causes of vehicle failure and can be used throughout fabrication, shipping, assembly, and pre-launch operations [63]. Other improvements to reliability in this area can be achieved by performing very detailed analyses of the flight trajectory. This analysis ultimately feeds into the go/no-go decision making process on the day of the launch.

The second category, programmatic environment, affects a vehicle throughout the entire life-cycle. This category includes any drivers related to planning and decision making. First and foremost is the effect of management style on reliability and safety. The management style relates to the rigor put into design, testing, fabrication, production, and assembly. Certain styles will require more control over many processes or require the use of various safety improvement methods [129]. In general, more control means the implementation of strict documentation and rigorous post test or post flight data analysis [94, 108]. Other management styles may include incentives for employees in order to ensure that the reliability and safety program is strictly followed. An example of such a program is the NASA Silver Snoopy award, which is given to employees or contractors who have made a significant impact on ensuring vehicle safety [30].

Another aspect of management style is the type of decision making that is performed throughout the program. Decision making can vary greatly from program to program, with some requiring very structured processes and others utilizing upper managers to make key decisions [30, 94]. Obviously, certain decisions during the vehicle's life-cycle will have a major impact on its reliability and safety. An example of the impact of decision making is the go for launch decision that was made for the Space Shuttle Challenger [94]. Management made the decision to launch the shuttle even though it was exposed to well below normal operating temperatures on the pad the morning of the launch. This decision ultimately lead to the loss of the crew, major investigations and redesigns, as well as a grounding of the shuttle for almost 3 years [27, 128]. Mitigation of these negative effects can be achieved by applying strict continuous risk management and decision making techniques [109, 120].

In addition to management style, developer experience is also a major factor [61, 91, 108]. Experience plays a key role in implementing the appropriate reliability and safety programs throughout the vehicle life-cycle. It also relates to the amount of knowledge possessed by the design team. At the beginning of the design phase, a more experienced team is expected to have more insight in regard to reliability and safety improvement strategies [108].

The final category of reliability and safety drivers for launch vehicles is vehicle architecture. Vehicle architecture refers to the physical description of the vehicle, including its subsystem types and specifications. This description includes variables such as: manned/unmanned, number of stages, number of engines, engine cycle, fuel type, oxidizer type, number of boosters, booster type, and redundancy.

One of the primary considerations for improving vehicle reliability using the vehicle architecture is engine-out design [84, 85, 88, 135]. A vehicle with engine out capability is expected to be more reliable because it can still deliver its payload to orbit if a single benign engine failure has occurred. A great example of this capability is the

Saturn V launch vehicle, which had engine out capability for its S-II second stage [88]. During the launch of the Apollo 13 mission in 1970, the S-II center engine was shutdown early leading to a deviation in the planned flight trajectory [127]. Due to the engine out capability, the payload was still successfully delivered to orbit and Apollo 13 was able to continue its mission to the moon.

In addition to the engine configuration, the engine cycle is also an important factor that affects vehicle reliability and safety. The engine cycle will effectively determine the complexity of the engine hardware that must be manufactured and successfully operated. For example, an expander cycle engine is considered to be far less complex than a gas generator cycle due to the inclusion of high pressure turbo pumps and associated controls [61]. This complexity relates directly to the number of components in the propulsion subsystem, and ultimately the number of potential failure modes that can be encountered. Increasing complexity is generally expected to decrease the reliability of the vehicle [63].

Another vehicle architecture parameter that affects reliability is propellant type. Each different propellant combination will have its own considerations for proper storage and use in a launch vehicle. The different types will introduce certain failure modes into the system that may be unique between propellants [24]. For example, the use of hydrogen may cause embrittlement issues, while material compatibility with liquid or gaseous oxygen may also be an issue [84]. The explosive properties of the propellants are also an important consideration in regard to crew safety in a catastrophic failure event [100].

Related to the propellant type selection is the tank configuration. Tank configuration refers to the design strategy, materials, and location of the tanks in relation to other vehicle components. The first consideration for propellant tanks is the tank locations, which stem from the initial design concept of the vehicle. A majority of launch vehicles utilize an “in-line” configuration in which the propellant tanks are stacked on top of one another. In this configuration considerations must be made in regard to the propellant feed lines, systems tunnels, and thermal management [91].

For other configurations, the location of the tanks in relation to other tanks or key systems is an important consideration. This is especially important in catastrophic failure events where a local explosion may lead to explosion of other tanks in the near vicinity [100]. The location of the tanks in relation to other vehicle components can also become an issue during nominal operations. For example, the thermal protection system of the Space Shuttle external tank (ET) was made up of light polyurethane foam, which often detached during launch [114]. Due to location of the ET immediately below the Space Shuttle orbiter, debris strikes to the orbiter were an additional failure mode introduced into the system. This failure mode surfaced during STS-107 when the orbiter Columbia's left wing was damaged by a foam impact at launch, ultimately leading to the loss of the orbiter during re-entry [114].

The descriptions of architecture parameters given above are only a small subset of the options that affect vehicle reliability and safety. Each of these architecture options introduce failure modes into the system and may cause negative interactions with other options. Consequently, as options are selected that define the baseline vehicle, certain failure modes will be "locked-in" to the system. This has an obvious effect on the vehicle reliability and safety during operations, as these failure modes could cause LOM or LOC.

An important observation can be made at this point. Vehicle architecture affects reliability and safety during operations, but it is decided upon during early design when a baseline vehicle is selected. Unless a great amount of money and effort is spent, the architecture will remain the same from conceptual design through operations. Therefore, architecture decisions are of utmost importance to the reliability and safety that a vehicle will achieve. This conclusion has been widely supported by authors in the launch vehicle design community [14, 49, 91, 159, 168].

1.2 Research Focus and Organization

Many authors support the importance of architecture selection on reliability and safety; however, these effects are not typically taken into account during early design. This is primarily because reliability analysis is considered to be a detailed design activity [21, 60, 129]. Therefore the focus of this thesis is on the inclusion of reliability and safety analyses during conceptual design. In order to make architecture decisions that improve reliability and safety, the designer must be able to quantify these effects prior to defining the baseline concept. The inclusion of reliability analysis upfront in the design process gives the designer more information on which to base architecture decisions. Ultimately, this will result in the selection of a vehicle that will have the highest probability of meeting the reliability and safety requirements.

In order to address the movement of reliability analyses from the detailed design phase into conceptual design a review of current techniques will be given first. Chapter 2 begins with a review of the generic design process for launch vehicles, which is followed by a discussion of requirements and guidelines for reliability analysis. The first two sections of Chapter 2 are therefore used to illustrate the analyses that are applied at certain points during the design cycle of a launch vehicle. Following this discussion, specific techniques for including reliability and safety considerations during architecture selection are identified. Observations are drawn from these existing techniques, which leads to the development of the specific research objective for this thesis in Section 2.4.

After defining the research objective, Chapter 3 presents the development of the ConcepTual Reliability Growth Approach for CompariSon of Launch Vehicle ArchiTectures (CONTRAST). Within Chapter 3 the method is developed step-by-step using a combination of research questions, literature review, and experimentation. A total of 6 research questions are presented in this chapter along with 3 experiments.

Chapter 4 presents a detailed description of each of the steps within the CONTRAST method. The goal of the chapter is to provide enough detail to allow the reader to reproduce and apply the method. At the end of Chapter 4 an example problem is presented, which acts as a validation of the method output. This problem illustrates the ability of the method to capture the reliability growth behavior of a previous launch vehicle.

Following the validation exercise, Chapter 5 presents an application of the CONTRAST method to a relevant vehicle design problem. Within this Chapter the requirements for research objective completion are verified using the output of the method. In addition, detailed discussion of the method output versus a previous state of the art tool is presented. Chapter 5 concludes with analysis of the future block upgrades for the vehicle of interest.

The final chapter begins to wrap up the thesis by presenting a summary of the findings from the literature review, experiments, and application problem. The second section in Chapter 6 presents the contributions stemming from the completion of this work. Finally, Section 6.3 identifies areas of interest for future research and extension of the CONTRAST method.

CHAPTER II

BACKGROUND

After narrowing the focus of this thesis to look at the effects of vehicle architecture on reliability and safety, another motivating question can be derived. This question will be used to guide the background literature survey, which leads to the derivation of the overall research objective of the thesis. The additional motivating question asks: what reliability and safety techniques are used throughout a typical launch vehicle design process? This question seeks to identify any current approaches for reliability and safety assessment of launch vehicles. While answering this question, the types of techniques as well as their applicability during specific steps in the design process will become apparent. This can ultimately be used to direct the development of the CONTRAST method.

2.1 Launch Vehicle Design Process

The motivating question identified above addresses the type and applicability of various reliability and safety techniques during launch vehicle design. In order to understand the application of these techniques, a review of the typical design process is necessary. This process can be broken in to four general phases: Pre-Conceptual, Conceptual, Preliminary, and Detailed [14, 43, 116].

The first phase, pre-conceptual, consists of design studies that examine various feasible vehicle concepts for general missions of interest. The primary purpose of this phase is to identify vehicle concepts from which new projects can be selected [116]. The identified vehicle concepts may also lead into advanced studies, which may extend for several years. Advanced studies focus on top-level system requirements, mission goals, and concept of operations [116]. Pre-conceptual design activities are usually performed continuously by concept study groups [116]. It is important to note that baseline vehicle architecture selection is typically performed at the end of the pre-conceptual phase or during the conceptual phase.

The second phase, conceptual design, includes a more detailed look at baseline mission concepts, mission requirements, and mission objectives. During this phase, activities become more formal and the emphasis is shifted towards optimality rather than feasibility [116]. More detailed analysis is performed using top-level sizing to produce estimates of performance, cost, technology development needs, and risk [14]. Using identified selection criteria, the number of feasible concepts being considered is narrowed as the design progresses towards a baseline. Thus, a typical result of the conceptual design process is the selection of a single baseline concept [14, 116].

The preliminary design phase is characterized by increased fidelity analysis of all significant subsystems [14]. During this phase the project level performance requirements are used to compile a complete set of system and subsystem design specifications for both flight and ground elements [116]. An evolving baseline is carried throughout the phase, which may encounter fundamental changes to its architecture or small refinements to the subsystem designs. At this point engineering test items may also be developed in order to derive data for evaluation of project risk or to demonstrate new technology [116]. The preliminary design phase ends with a preliminary design review (PDR) in which all analysis and design work is used to generate the final design-to specifications for the system. After the PDR, any design changes are expected to represent successive refinements, with no fundamental changes [116].

The fourth phase, detailed design, provides complete specification of all hardware and software of the system [14, 116]. These specifications will allow for the production of test articles as well as generation of detailed analyses to continue in verifying the performance of the system. The analyses and tests are performed in order to increase the confidence that the design will function as expected [116]. In addition to refining the design, plans for manufacturing, integration, operations, and support are also considered. The final product of the detailed design phase is a full definition of the system to be fabricated, including any relevant project plans going forward [14, 116]. At this point the vehicle is ready to proceed into production and operations.

2.2 Launch Vehicle Reliability and Safety Programs

Existing reliability and safety techniques can now be mapped to the generic design process described above. In order to identify the reliability and safety methods that are in use today, a literature search was focused on three primary entities that control much of the launch vehicle domain. The three primary entities are the National Aeronautics and Space Administration (NASA), the U.S. Department of Defense (DoD), and the European Cooperation for Space Standardization (ECSS). It is expected that almost any new launch vehicle program in the world would be governed by the requirements and guidelines laid out by these organizations.

In the United States, the DoD has published many requirements documents and military standards that are used by any contractor seeking to carry government payloads. A few examples include United Launch Alliance (ULA), Orbital Sciences, or Alliant Techsystems (ATK) [126, 160]. Commercial companies seeking to produce manned spacecraft, such as SpaceX or Sierra Nevada, will follow various NASA guidelines and requirements in addition to DoD military standards. In Europe, the ECSS produces guidelines, standards, and requirements documents that are used by the European Space Agency [55]. Due to the widespread use of the documents produced by these three entities, their requirements and standards represent a full list of safety and reliability methods that are used today.

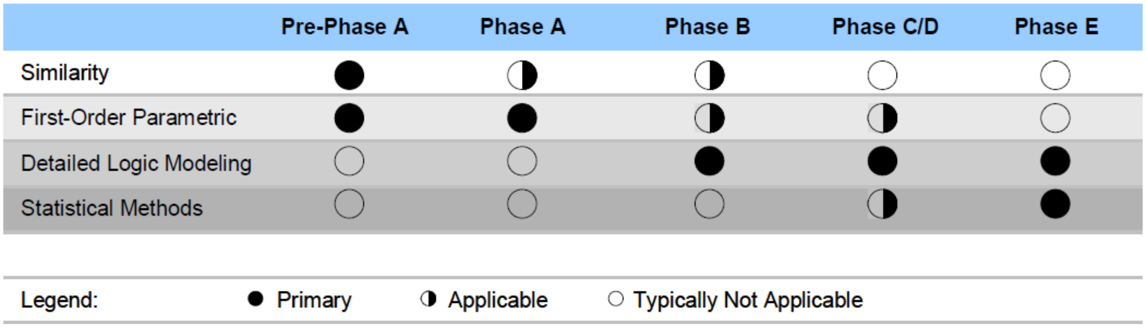
2.2.1 NASA Requirements and Guidelines

The design process as defined by NASA follows the general phases outlined in Section 2.1 very closely. The four design phases identified by the NASA Systems Engineering

Handbook are Pre-Phase A, Phase A, Phase B, and Phase C [116]. Pre-Phase A refers to the pre-conceptual design phase, where broad studies of vehicle concepts are performed. Phase A continues conceptual design, ultimately leading towards the selection of a final baseline concept. Phase B moves into the preliminary design phase and may also constitute production of test articles and mockups. Phase C represents the detailed design phase, in which the end product specifications are finalized in preparation for production.

The NASA Standard 8729.1, Planning, Developing, and Maintaining an Effective Reliability and Maintainability Program states that the reliability program should be tailored to each specific program in order to capture the effects deemed most important to project success [112]. In addition to a tailored reliability program, the NASA handbooks call for a proactive approach for risk reduction, which includes two components; Risk-Informed Decision Making (RIDM) and Continuous Risk Management (CRM) [117, 120]. Continuous risk management calls for the application of techniques to track and reduce risk throughout the system's life-cycle. The RIDM process is one of the techniques applied in CRM [109]. In the RIDM process, three steps are used to support decision making; Identification of alternatives, Risk Analysis of Alternatives, and Risk-Informed Alternative Selection [109]. The RIDM approach is not a reliability and safety assessment technique, but its second step can include such methods.

The RIDM handbook provides an outline of applicability of some specific reliability and safety techniques during the various design phases. In Figure 1, a table from the RIDM handbook is displayed, which identifies the applicability of various reliability and safety techniques throughout the design phases [109].



	Pre-Phase A	Phase A	Phase B	Phase C/D	Phase E
Similarity	●	◑	◑	○	○
First-Order Parametric	●	●	◑	◑	○
Detailed Logic Modeling	○	○	●	●	●
Statistical Methods	○	○	○	◑	●

Legend: ● Primary ◑ Applicable ○ Typically Not Applicable

Figure 1: Applicability of various reliability and safety methods during design

As illustrated in the figure, a list of four types of reliability and safety assessment techniques are mapped against five life-cycle phases. The first four phases, Pre-Phase A to Phase C represent the design phases of the program. Two additional phases are included in the figure, Phase D and Phase E. These represent system assembly, integration, and test, and system operations, respectively. Through the design phases the figure shows that only the first three types of methods are applicable, with the fourth method being potentially applicable at the conclusion of Phase C.

The first identified type is similarity, which utilizes comparison and extrapolation to estimate reliability and safety [109]. Similarity methods often use operational data from past programs that is determined to be technically representative of the new system [109]. This data can then be subjectively adjusted depending upon the assumed differences between the complexities of the systems. After adjustment, the data is assumed to represent the expected reliability of the new system.

The second assessment type is first-order parametric. First-order techniques are primarily used during the conceptual and pre-conceptual design phases. This type is similar to the similarity method in that it utilizes data from past programs. In this estimation technique simple mathematical expressions are derived based upon the historical data, which are then used to estimate the reliability of the new system. An implicit assumption of this technique is that the same factors that shaped the reliability and safety in the past will affect the system being assessed [109].

Detailed logic modeling is identified as the third assessment type. These techniques are generally applicable during Phase B, which is preliminary design. Detailed logic modeling involves “top-down” development of scenario-based or discrete event logic models [109]. Specific examples of logic models include fault tree analysis, event trees, and reliability block diagrams. Detailed simulation or testing can be used in these techniques to develop pdfs for quantification of the model [109]. Typically these pdfs give the analyst the probability that certain undesired top-level events, such as LOC or LOM, will occur.

2.2.2 DoD Requirements and Guidelines

The design process as defined by the U.S. Department of Defense Military Standards varies from the general process discussed in Section 2.1. In MIL-STD-1543B, Reliability Program Requirements for Space and Launch Vehicles, four acquisition phases are laid out [43]. These phases are the conceptual phase, demonstration and validation phase, full-scale engineering development phase, and the production phase. The first two phases will be discussed in this section because they correspond to the major design activities. The engineering development phase and the production phase refer to manufacturing and operation of the system, respectively, thus they will not be included in the discussion below.

The conceptual phase corresponds to the pre-conceptual and conceptual design phases discussed in Section 2.1. This phase involves identification and exploration of vehicle concepts that have the potential to satisfy a validated operational need [43]. During this phase the reliability program objectives are to derive values of reliability characteristics and to refine the quantitative system reliability goals based on system level trade studies [43]. For this phase, functional level failure modes and effects analysis is identified as the primary method of reliability assessment [43]. As defined by MIL-STD-1619A, Procedures for Performing a Failure Mode, Effects, and Criticality Analysis, the process of FMEA also includes the creation of a physical or functional block diagram of the system being analyzed [40].

The second phase, demonstration and validation, focuses on refining selected candidate concepts by performing more extensive studies and analyses or developing test hardware [43]. This phase corresponds to the preliminary design phase and the early stages of the detailed design phase discussed in Section 2.1. The goal of this phase is to demonstrate the feasibility of one or more of the candidate concepts before entering into the full-scale development phase [43].

During the demonstration and validation phase, a few different reliability and safety methods are identified for use. The first is the implementation of a formal failure reporting and corrective action system [43]. This method is implemented only in the case where physical developmental testing will be carried out on components within the system. Additional techniques such as FMEA and generic system level math models are identified for use during the validation phase. The mathematical models discussed in the military standards are equivalent to the first-order parametric models that were discussed in the NASA literature. These models rely on historical data and similarity to past systems in order to produce reliability estimates for new vehicles [42].

Although less documentation is available, the military standards identify many techniques already contained within the NASA documentation. These techniques include failure modes and effects analysis, reliability block diagrams, hazard analysis, stress-strength analysis, parts count method, and similarity method.

2.2.3 ECSS Requirements and Guidelines

The design process as defined by the European Co-operative for Space Standardization varies from the general process described in Section 2.1. The ECSS requirements document, “Space Product Assurance: Safety”, illustrates a five phase design process including utilization [53]. As with the DoD requirements discussed above, only the early design phases will be included in this section. Thus, the detailed definition, production, and qualification testing phase and the utilization phase will not be included.

The first phase of the design process, mission analysis and needs identification, corresponds to the pre-conceptual phase from Section 2.1. During this phase, the goal of the reliability and safety program is to support the identification of sources of safety risk as well as the performance of preliminary trade-off analyses between alternative system concepts [53]. The ECSS requirements specifically call for the use of preliminary hazard analyses and comparative risk assessment of the concept options [53]. The comparative risk analysis from these requirements is equivalent to the similarity methods called out by the NASA and DoD requirements. For similarity comparisons, the ECSS also presents a methodology for the proper selection of reliability data [54].

The second phase of the design process corresponds to the conceptual design phase discussed in Section 2.1. This phase is the feasibility phase through which design alternatives are pared down until a single baseline concept remains [53]. During the feasibility phase, the primary goal of the reliability and safety program is to support trade-off analyses in arriving at a concept that has acceptable safety risk considering the project constraints [53]. This goal is met through the continued use of hazard analyses and similarity techniques.

The third phase of the ECSS design process is the preliminary definition phase, which corresponds to the preliminary design phase discussed in Section 2.1. During the preliminary definition phase the design process proceeds with a single baseline concept and further details in regard to the system design and operations are developed. The goal of the reliability and safety program during this phase is to support the detailed optimization of the system design and operations [53]. This goal is accomplished through the use of the same techniques from the previous phases and other techniques. The additional reliability and safety techniques used during the preliminary definition phase are fault tree analysis and failure modes and effects analysis.

Similar to the NASA and DoD requirements and guidelines documents a very common set of reliability and safety techniques are identified by the ECSS requirements. These techniques include similarity method, failure modes and effects analysis, fault tree analysis, and hazard analysis, all of which have been identified previously.

2.2.4 Observations

After a review of the guidelines and requirements documents from three major entities, two primary observations can be derived. First, it was observed that many reliability techniques exist for application throughout the vehicle design process. The documentation from NASA, DoD, and ECSS share a common set of reliability and safety analysis tools, which range from qualitative techniques such as FMEA to detailed quantitative techniques such as discrete event logic modeling. The list of tools identified within the documentation includes; FMEA, hazard analysis, fault trees, reliability block diagrams, similarity, parts count, stress-strength, and detailed logic modeling. This ultimately shows that some form of reliability analysis can be performed at any point in the design process.

The second observation from the review refers to the goals of the reliability analyses applied throughout the design cycle. Although reliability analysis can be performed during each phase, the focus of the analyses changes drastically as the design progresses. As noted in the NASA standards, detailed logic modeling and statistical methods are identified as the primary tools for Phase B or later, which is after the baseline vehicle selection has occurred. The purpose of these tools is to pinpoint a more accurate estimate of the expected reliability of the vehicle. As stated by the NASA Systems Engineering Handbook, during the latter phases, detailed techniques are used to verify that the design is meeting its risk and reliability goals [116].

The reliability analysis goals during the latter design phases are mirrored in the DoD and ECSS standards as well. During the latter phases of the DoD design process, the reliability models are said to progress from the subsystem level down to the component and part levels as details become more firm [43]. The ECSS documentation notes that the goal of the reliability and safety program during the latter phases of design is to support detailed optimization of the system [53].

Considering the early phases of design, the techniques identified by NASA, DoD, and ECSS serve a different purpose. Due to the lack of design knowledge at this point in the process, it is very difficult to pinpoint an exact value for the system reliability. Instead, the key contribution of the reliability analysis is to make the designers aware of impacts of their decisions on system reliability [116]. Ultimately, the focus of the techniques applied during early design is to capture the relative effects of decisions on system reliability. Although the techniques cannot give an exact estimate for the system reliability at this point in time, the ability to compare between system options adds value to the design process.

The goals of reliability analyses during early design are therefore to support the decision maker during initial system definition. This goal links directly to the discussion from Chapter 1, which identified the importance of architecture selection on the eventual reliability and safety of the system. A second motivating question can be derived at this point, which asks: what techniques exist that utilize reliability and safety as figures of merit for selection of a baseline vehicle architecture?

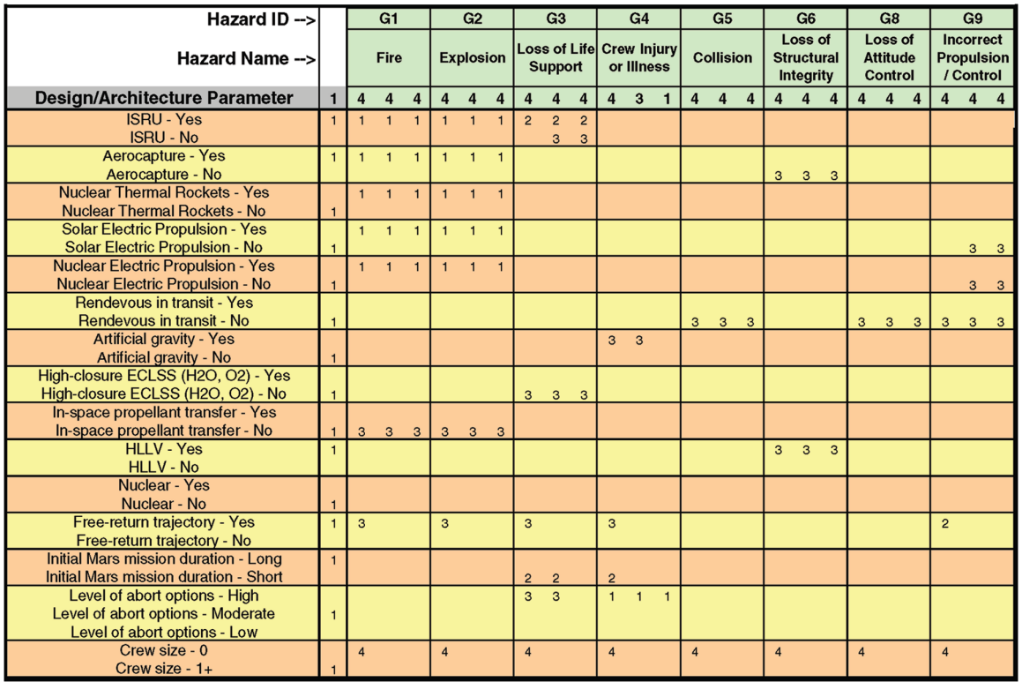
2.3 Reliability and Safety Based Architecture Selection

The second motivating question seeks to identify specific existing techniques that can be used to infuse reliability and safety considerations into baseline system selection. The ability to capture these effects on reliability and safety represents the primary approach for improving reliability and safety in the area of vehicle architecture. In all, four existing approaches were identified that address vehicle architecture selection. The qualitative techniques will be discussed first, followed by the quantitative approaches.

2.3.1 Qualitative Methods

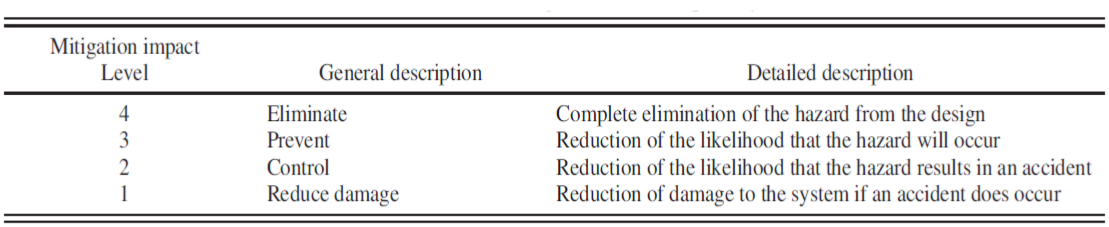
The first two methods to be reviewed use a hazard analysis based approach. They utilize qualitative information collected from subject matter experts to produce ranked lists of architectures deemed to be the “safest”. The first method, Hazard-based Safety/Risk analysis, was proposed by Dulac and Leveson in 2009 [49]. The proposed methodology begins by identifying all system level hazards and their associated severities. The identification step is achieved through the completion of hazard worksheets by subject matter experts. After the system level hazards have been identified, the next step is to identify any mitigating strategies for each hazard as well as the associated impact of each strategy. These impacts are based upon a standard four point scale, which can be seen in Figure 3 [49].

For each of the identified architecture options, the impact scores are mapped across the identified hazards utilizing a simple table. Figure 2 shows an example section of the table used by Dulac and Leveson [49]. After populating the table, the various architecture alternatives can be evaluated. First, by selecting an option in each category an impact factor for each hazard is determined. For each hazard a relative residual risk index is calculated, which is based upon the ratio between the selected option impact factor and the overall maximum possible value of the impact factor for the given hazard. A relative severity index is then created for each hazard by multiplying the relative residual risk index by the square of the given hazard's severity. After generating relative severity indices for each of the hazards in the table, the overall residual safety risk metric is obtained. This metric is calculated using a weighted average of the relative severity indices across all the hazards. After repeating this process for all the possible combinations of architecture options, a ranked list can be created. The ranked list represents the architectures that are expected to be at the least risk of encountering the identified hazards.



Hazard ID -->		G1			G2			G3			G4			G5			G6			G8			G9		
Hazard Name -->		Fire			Explosion			Loss of Life Support			Crew Injury or Illness			Collision			Loss of Structural Integrity			Loss of Attitude Control			Incorrect Propulsion / Control		
Design/Architecture Parameter	1	4	4	4	4	4	4	4	4	4	4	3	1	4	4	4	4	4	4	4	4	4	4	4	4
ISRU - Yes	1	1	1	1	1	1	1	2	2	2															
ISRU - No									3	3															
Aerocapture - Yes	1	1	1	1	1	1	1																		
Aerocapture - No																	3	3	3						
Nuclear Thermal Rockets - Yes		1	1	1	1	1	1																		
Nuclear Thermal Rockets - No	1																								
Solar Electric Propulsion - Yes		1	1	1	1	1	1																		
Solar Electric Propulsion - No	1																							3	3
Nuclear Electric Propulsion - Yes		1	1	1	1	1	1																		
Nuclear Electric Propulsion - No	1																							3	3
Rendevous in transit - Yes																									
Rendevous in transit - No	1													3	3	3				3	3	3	3	3	3
Artificial gravity - Yes											3	3													
Artificial gravity - No	1																								
High-closure ECLSS (H2O, O2) - Yes																									
High-closure ECLSS (H2O, O2) - No	1							3	3	3															
In-space propellant transfer - Yes																									
In-space propellant transfer - No	1	3	3	3	3	3	3																		
HLLV - Yes	1																3	3	3						
HLLV - No																									
Nuclear - Yes																									
Nuclear - No	1																								
Free-return trajectory - Yes	1	3			3			3			3												2		
Free-return trajectory - No																									
Initial Mars mission duration - Long	1																								
Initial Mars mission duration - Short								2	2		2														
Level of abort options - High								3	3		1	1	1												
Level of abort options - Moderate	1																								
Level of abort options - Low																									
Crew size - 0		4			4			4			4			4			4			4			4		
Crew size - 1+	1																								

Figure 2: Sample hazard mitigation table [49]



Mitigation impact Level	General description	Detailed description
4	Eliminate	Complete elimination of the hazard from the design
3	Prevent	Reduction of the likelihood that the hazard will occur
2	Control	Reduction of the likelihood that the hazard results in an accident
1	Reduce damage	Reduction of damage to the system if an accident does occur

Figure 3: Hazard mitigation impact scale [49]

The method proposed by Dulac and Leveson provides the analyst with a traceable way to incorporate system safety into early design. This method offers the benefit of being able to directly compare architectures before a selection is made. It is also very simple to perform as long as subject matter experts are available to give hazard information.

The primary issue with this method is that it does not produce quantitative estimates for the architecture safety. This becomes a problem when trying to compare two architectures from the ranked list. For example, consider an analyst trying to decide between architecture #1 and architecture #2 from the list. There is no way for the analyst to know how much safer option #1 is than option #2. In this sense the analyst will not know the relative cost and benefit of selecting architecture #1 over architecture #2. Another issue with this method is the reliance on subject matter expert input. Although SME input in itself is not a weakness, it does limit the number of architectures that can be evaluated. The SME input portion of this method could take a very long time to complete if many different architecture options were being considered.

The second qualitative method was proposed by Fabisinski and Maples [56]. This method utilizes a Source-Taxonomy based approach for risk identification in space architectures. The analysis begins with the creation of a taxonomy of possible risk sources for the program. Based upon this taxonomy a questionnaire is created for each discipline expert to answer. The purpose of the questionnaire is to identify which of the risk sources from the taxonomy are applicable to some aspect of the discipline expert's field. Ultimately, the questionnaire reveals which risks are related to each risk source.

After performing the questionnaire, the resulting list of risks is vetted by the analyst. Any risks considered as very low likelihood are candidates for elimination from the list. The remaining risks are then given a likelihood and consequence value by the discipline experts. After assigning likelihoods and consequences an expected risk value is calculated for each risk in the list. This score is simply the product of the likelihood and consequence values. In order to evaluate the aggregate risk score for a candidate architecture, the expected risk values from all the risks applicable to the specific architecture are used. This score is used to produce a major risk report for the given architecture, which allows the analyst to produce a ranked list of alternatives.

The method proposed by Fabisinski and Maples has many of the same strengths and weaknesses as the method proposed by Dulac and Leveson. The first benefit of using this method is the ability to assess the effects of various risk factors on the overall system architecture. This allows analysts to identify potential weak points in the design or concept of operations for the design reference mission. The method has also been incorporated into an easy to use tool called the Advanced Concepts Evaluating Risk Tool (ACERT), which has been tested by the Advanced Concepts Office at Marshall Space Flight Center (MSFC) [56].

On the downside, the ACERT tool requires the same amount of SME input as Dulac and Leveson's method. This can severely limit the number of architectures that can be evaluated due to the large time commitment for developing the taxonomy and performing the questionnaires. The issue of direct comparison also comes into play with this method. Since the likelihood and consequence values are entered as probabilities the analysts can get an idea of the relative increase or decrease in reliability and safety that may occur by switching between architectures. However, the likelihood and consequence values are still subjective in nature, which may cause the comparisons to be suspect.

Overall, the qualitative methods available for evaluation of architecture effects on reliability and safety share the same strengths and weaknesses. They allow for the consideration of reliability and safety during early design, but are ultimately limited in applicability due to their reliance on qualitative and subjective data. The next section will discuss the existing quantitative methods for reliability and safety driven architecture selection.

2.3.2 Quantitative Methods

The first example of a quantitative method for evaluation of reliability and safety for architecture selection was presented by Krevor in 2007. Krevor proposed a methodology to link cost and reliability, which accounted for a few different vehicle architecture parameters [88]. In his study, the method was applied to the reliability prediction of the Saturn V and SLS launch vehicles. Krevor's reliability assessment method utilized fault tree analysis to produce estimates for probability of LOM. The fault trees were generated based upon the selection of two primary architecture options redundancy and engine out. It is important to note that Krevor did not include any of the other architecture options, as an assumption was made that the preliminary down selection of the baseline concept was previously performed. Even though his method did not include all architecture options it does demonstrate the applicability of FTA to architecture evaluation.

The primary strength of Krevor's method was in the rapid evaluation of the candidate architectures. Automatic generation of the fault trees allowed Krevor to evaluate all of the options in the defined architecture space. The main issue with this method however, is related to the lack of a full architecture evaluation. As was mentioned previously, most of the architecture options were assumed away, leaving only two to play with. This severely limited the architecture space that was evaluated.

The second qualitative method example is the Flight-oriented Integrated Reliability and Safety Tool (FIRST), which was developed by SAIC [16, 153]. This tool utilizes preliminary vehicle descriptions along with historical data from previous launch vehicle elements to produce reliability and safety estimates in the form of probability of LOM or LOC [16, 153]. Due to its use of historical data the FIRST tool can be classified as a type of similarity method. To produce a reliability or safety estimate for a new vehicle, the similarity of that vehicle to the data in the database is assessed. The analysts are also able to adjust the data accordingly based upon predictions that reflect future testing or design improvements. The FIRST tool displays the reliability and safety output as a probability distribution for LOM or LOC, respectively. Figure 4 illustrates distributions produced by FIRST for the NASA Exploration Systems Architecture Study in 2005 [153].

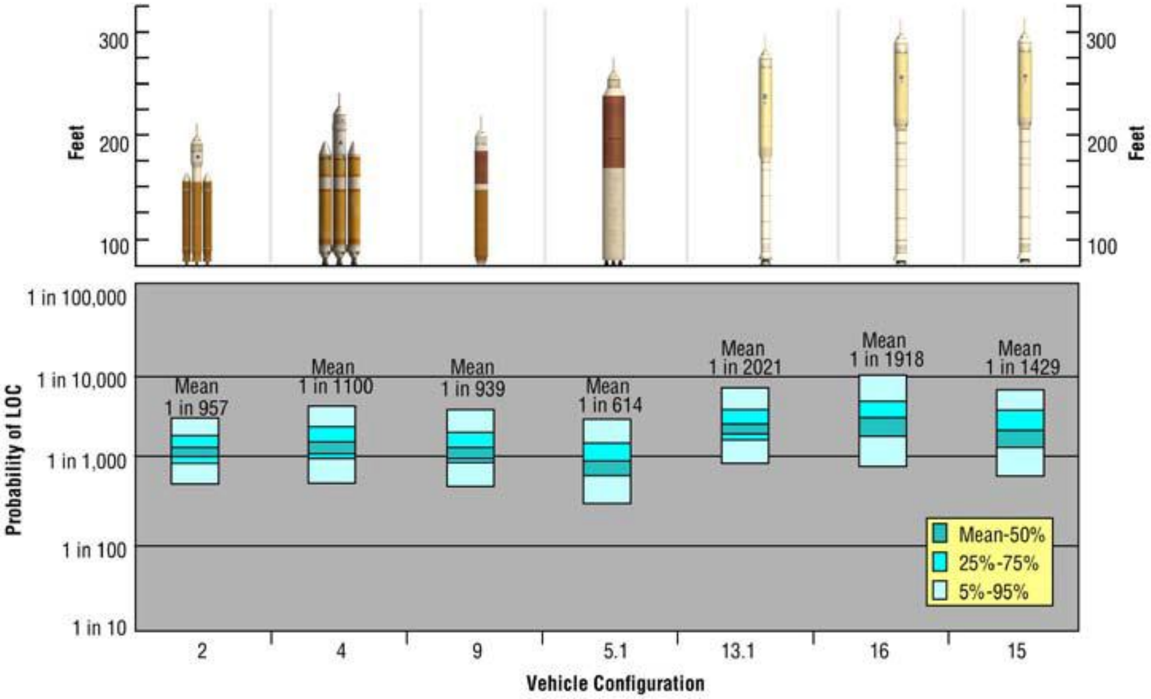


Figure 4: Probability of LOC output from FIRST tool

As can be seen in the figure, direct comparisons between vehicle concepts can be made. The primary benefit of the FIRST tool is that it produces quantitative predictions of both reliability and safety. This allows the analyst to assess the relative difference in expected LOM and LOC between two concepts. It also allows the analyst to rapidly assess whether or not a given concept will be able to meet the reliability and safety requirements for the program. The FIRST tool gives analysts the ability to rapidly assess many different vehicle architectures in a quantitative manner.

It is important to note that the FIRST tool uses the assumption that all of the vehicles have reached a "mature" state. This means that the vehicles have already gone through a reliability growth period. Thus the output distributions represent the maximum expected reliability or safety of the vehicle. The mature assumption is the largest weakness in the FIRST tool, which can lead to a major consequence.

The reliability growth and maturation process of launch vehicles has been well documented [61, 63, 78, 101, 108]. For some vehicles this process may take only a dozen flights, yet for others it may take on the order of 100's to reach maturity [61]. For this reason the mature reliability and safety distributions from FIRST are an incomplete picture of the expected performance of the vehicles. If two vehicles have very similar reliability and safety distributions, there is not a way for the analyst to ensure that one or the other will actually reach the required reliability and safety.

For example, consider a new program that is expected to fly on the order of 25 flights before retirement of the vehicle. Using the output of the FIRST tool a vehicle is selected that has a very desirable reliability distribution, which is well above the required reliability for the program. However, if this vehicle requires more than 100 flights to reach maturity, there is no guarantee that the required reliability will be achieved. At the last planned flight of the program, the reliability distribution may be much different than the mature distribution produced by FIRST. This example identifies a major consequence of using mature estimates to select an architecture. The use of these estimates without an evaluation of the expected time to maturity may lead to the selection and commitment to a vehicle that will not meet the reliability and safety requirements in the future.

The two quantitative methods reviewed in this section represent the current state of the art in reliability and safety driven architecture selection. These methods give the analyst the ability to assess architecture trades during early conceptual design.

However, many weaknesses in these methods have been identified. These weaknesses helped lead to the conclusion that improvement is needed in the area of reliability and safety driven architecture selection. From this conclusion, the specific research objective for this thesis can be derived.

2.4 Research Objective

The primary motivation for this research is the gap between current launch vehicle reliability and safety and the future goals set forth by the NASA Astronaut Office. It has been identified that vehicle architecture has a large effect on reliability and safety, making architecture decisions crucial to the success of new programs. Due to the fact that architecture decisions are made during early conceptual design, they are often made with very limited knowledge of the final design. Therefore, it is important to evaluate the effects of architecture decisions on reliability and safety in order to make a more informed selection of the initial concept. In the previous section existing methods for evaluating reliability and safety of conceptual launch vehicles were identified, however they lack the ability to fully support decision making at this point in the design phase. Therefore, the goal of this thesis is to improve upon current reliability and safety assessment methods for early conceptual design. Specifically, improvements will be made to help facilitate architecture decisions during vehicle concept selection, which is reflected in the overall research objective.

Research Objective

To formulate and implement a method that will quantitatively capture launch vehicle architecture effects on reliability and safety, in order to facilitate more informed decision making during early conceptual design.

In order to meet the overall research objective, requirements were derived whose completion will signify successful achievement of the objective. These requirements were derived based upon the identified weaknesses of the current techniques and are enumerated below. First, the qualitative techniques do not give an accurate picture of the relative difference between two concepts. This leads to the first requirement, which states that quantitative estimates are desired.

The second requirement stems from some of the shortcomings of the quantitative techniques in Section 2.3.2. Specifically, the output of the state-of-the-art FIRST tool provides only a mature reliability or safety estimate for the vehicles. This distribution does not capture the full picture of the risk of meeting the future requirements and does not provide a large amount of information with which to compare concepts.

The final requirement stems from the ability of the existing techniques to evaluate large architecture spaces. The qualitative techniques are fairly limited because they require a large amount of subject matter expert input for each architecture option. A trade space with tens of thousands of concepts would therefore be impractical using these techniques. The quantitative techniques can assess more vehicles than the qualitative techniques; however, issues may arise when considering new and novel concepts. For this reason the third requirement calls for the method to have enough flexibility to include novel concepts, while simultaneously enabling the analysis of large architecture spaces.

1. The method shall produce quantitative estimates for reliability and/or safety of the given launch vehicle concepts

2. The method shall have sufficient accuracy to enable comparison between unique but similar concepts

3. The method shall be flexible enough to evaluate any potential launch vehicle concept within the defined architecture space

With the research objective and requirements for objective completion in place, the derivation of the method can proceed. First, it is important to clarify the primary goal of the method itself when applied during early design. As stated in Section 2.2, availability of design information during the conceptual phase severely limits the reliability and safety analysis that can be performed. Due to this restriction the analyses at this point in the design process are focused on supporting trade-offs as the design progresses. The CONTRAST method developed in the following chapter is therefore aimed at supporting such trade-offs for initial baseline vehicle selection. Thus the output of the method is more concerned with capturing the effects of architecture options relative to one another than estimating the exact reliability of each specific vehicle. During conceptual design an exact estimate cannot be expected, however, the ability to determine relative differences in reliability and safety between concepts will be a valuable tool for the designer.

CHAPTER III

METHOD DEVELOPMENT

The development of the research objective for this thesis identified the need for improvement in the capture of architecture effects on reliability and safety during early design. In the previous section the ultimate goal of the method was stated, which is to support trade-offs between vehicle concepts using reliability and safety as figures of merit. The method therefore represents an additional decision support tool for use during conceptual design. To begin the development of the method a generic set of steps will first be defined using comparisons to existing decision-making processes. These generic steps will serve as a backbone for the derivation of the method.

3.1 Solution Approach

The first decision-making process to consider is NASA's Risk-Informed Decision Making (RIDM), which was referenced in Section 2.2. This process is a logical starting point because it relates directly to the assessment of risk within vehicle programs. The goal of the process is to provide support for key decisions including design options, source selection in major procurements, or budget allocation [109]. The RIDM process contains three primary parts, each with two steps. An illustration of the RIDM process can be seen in Figure 5.

The first part of the RIDM process is to define the alternatives. This part involves two steps, the first of which requires an understanding of the expectations and program measures of performance. The second step represents the identification of all possible alternatives for the problem of interest.

The second part of RIDM is to analyze the alternatives identified within part 1. The first step is to determine the methodologies to be used in the analysis. After selecting the analysis approach the second step within this part assesses each of the alternatives.

The final part of the RIDM process involves risk-informed alternative selection. The first step within this part requires an evaluation of the performance of the alternatives versus the program measures of performance from part 1. Finally, through deliberation and assessment of the results an alternative is selected and the rationale is recorded.

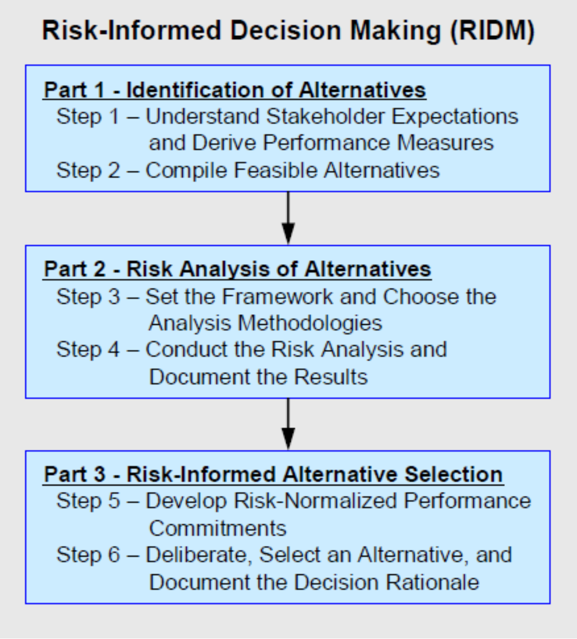

Risk-Informed Decision Making (RIDM)
Part 1 - Identification of Alternatives
Step 1 – Understand Stakeholder Expectations and Derive Performance Measures
Step 2 – Compile Feasible Alternatives
Part 2 - Risk Analysis of Alternatives
Step 3 – Set the Framework and Choose the Analysis Methodologies
Step 4 – Conduct the Risk Analysis and Document the Results
Part 3 - Risk-Informed Alternative Selection
Step 5 – Develop Risk-Normalized Performance Commitments
Step 6 – Deliberate, Select an Alternative, and Document the Decision Rationale

Figure 5: NASA Risk-Informed Decision Making process [109]

Although the RIDM process is specific to risk assessment of alternatives, it maps well to more generic processes. First, the three steps of the RIDM process mirror the three steps of the Concept Exploration and Refinement (CER) approach defined by the DoD [163]. Within CER three major sub-processes are defined; characterization of the trade space, characterization of the alternatives, and analysis of the alternatives [163]. The first two steps of the CER process map directly to part 1 of the RIDM process. The second is in line with part 2 of RIDM, which performs the analysis of alternatives.

The RIDM process can also be mapped to a more detailed but generic design decision support process; the Georgia Institute of Technology Integrated Product and Process Development approach (IPPD). A graphic overview of the IPPD methodology is shown in Figure 6. The approach was originally developed to evaluate technology for affordability [147]. However, the center column represents a generic top-down decision support process that can be applied to any problem of interest.

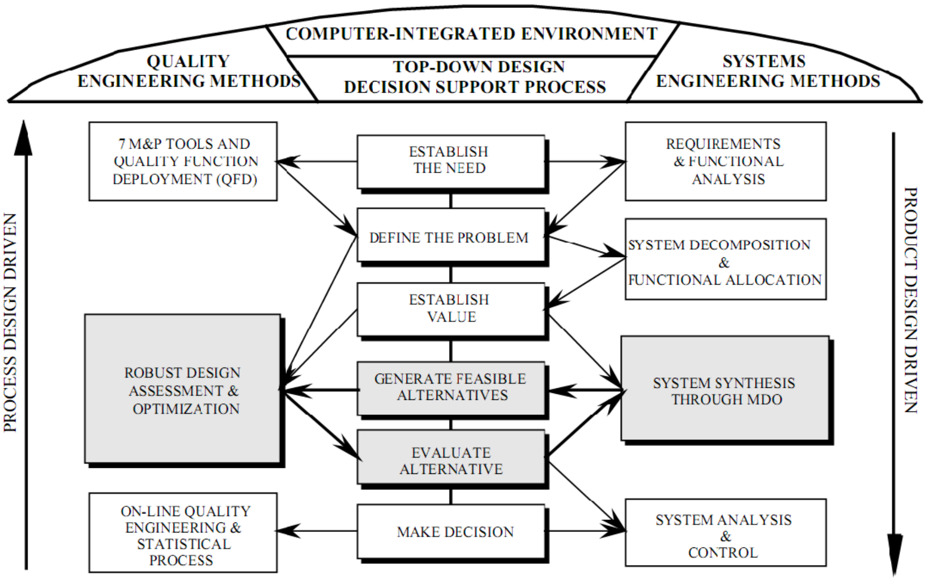

COMPUTER-INTEGRATED ENVIRONMENT
QUALITY ENGINEERING METHODS
TOP-DOWN DESIGN DECISION SUPPORT PROCESS
SYSTEMS ENGINEERING METHODS
PROCESS DESIGN DRIVEN
PRODUCT DESIGN DRIVEN
7 M&P TOOLS AND QUALITY FUNCTION DEPLOYMENT (QFD)
ESTABLISH THE NEED
REQUIREMENTS & FUNCTIONAL ANALYSIS
DEFINE THE PROBLEM
SYSTEM DECOMPOSITION & FUNCTIONAL ALLOCATION
ESTABLISH VALUE
ROBUST DESIGN ASSESSMENT & OPTIMIZATION
GENERATE FEASIBLE ALTERNATIVES
SYSTEM SYNTHESIS THROUGH MDO
EVALUATE ALTERNATIVE
ON-LINE QUALITY ENGINEERING & STATISTICAL PROCESS
MAKE DECISION
SYSTEM ANALYSIS & CONTROL

Figure 6: Georgia Tech IPPD Methodology [148]

The first part of the RIDM process can be mapped directly to the top four boxes within the center column of the IPPD methodology. The first step in part 1 of RIDM requires an understanding of the stakeholder expectations, which can be considered the same as the “Establish the Need” step within IPPD. The following steps of IPPD, “Define the Problem” and “Establish Value”, correspond to the derivation of the performance measures within RIDM. The fourth step in the center column of the IPPD methodology is “Generate Feasible Alternatives”, which is the final step within the first part of RIDM. The remaining two boxes within the central top-down decision support process of IPPD map directly to the second and third parts of RIDM. The "Risk Analysis of Alternatives" part in RIDM corresponds to "Evaluate Alternatives", while the final "Risk-Informed Alternative Selection" part maps to "Make Decision".

Considering the simple setup of RIDM and the generic nature of IPPD a generic three step process can be derived. This process represents a generic reliability and safety based decision approach, which will be used to guide the method development. Figure 7 shows a mapping of the RIDM and IPPD steps to the generic process.

First, the Establish the Need, Define the Problem, Establish Value, and Generate Feasible Alternative steps will be combined into a single Problem Definition step. For the purpose of reliability assessment of vehicle architectures, this step will correspond to the identification of architecture options as well as the metric of interest such as LOM, LOV, or LOC. The output of this step will be feasible vehicle architectures to be analyzed.

The Evaluate Alternatives and Risk Analysis of Alternatives steps within IPPD and RIDM, respectively, will also combine into a single Reliability and Safety Analysis step for the CONTRAST method. This generic step will house the designated approach for evaluating the reliability and safety of the given vehicle concept. Ultimately, this analysis approach will be determined based upon the output that is desired for successfully performing architecture comparisons. The final Make Decision step will therefore be stated as Architecture Comparison. The ultimate goal of this step is to identify potential baseline concepts that will have the greatest probability of meeting the reliability and safety requirements of the program.

A generic three step reliability and safety based decision-making approach has now been defined. These steps will be used to help guide the development of the CONTRAST method. A transposed version of the generic process can be seen in Figure 8, which will be used to trace the progress of the method development through the remainder of the chapter.

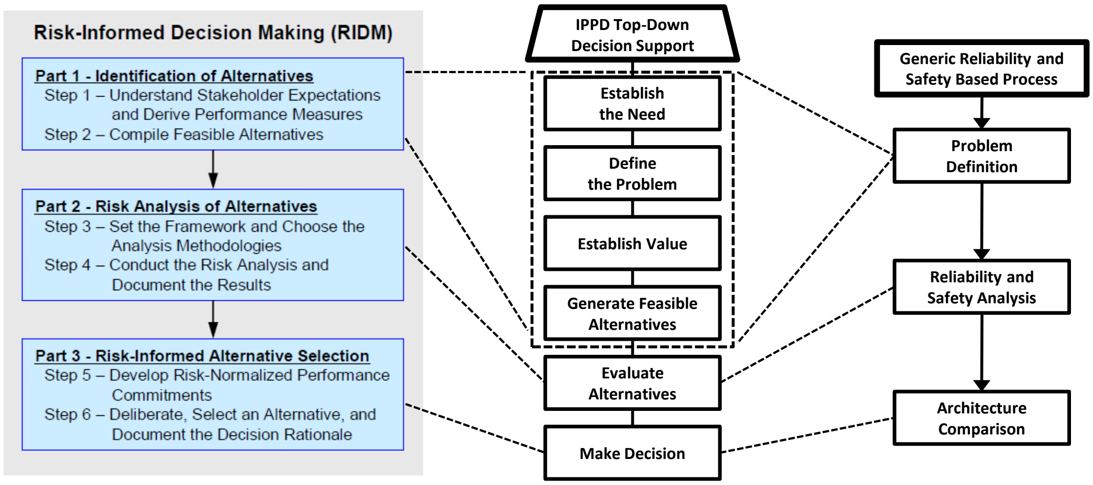

Risk-Informed Decision Making (RIDM)
Part 1 - Identification of Alternatives
Step 1 – Understand Stakeholder Expectations and Derive Performance Measures
Step 2 – Compile Feasible Alternatives
Part 2 - Risk Analysis of Alternatives
Step 3 – Set the Framework and Choose the Analysis Methodologies
Step 4 – Conduct the Risk Analysis and Document the Results
Part 3 - Risk-Informed Alternative Selection
Step 5 – Develop Risk-Normalized Performance Commitments
Step 6 – Deliberate, Select an Alternative, and Document the Decision Rationale
IPPD Top-Down Decision Support
Establish the Need
Define the Problem
Establish Value
Generate Feasible Alternatives
Evaluate Alternatives
Make Decision
Generic Reliability and Safety Based Process
Problem Definition
Reliability and Safety Analysis
Architecture Comparison

Figure 7: Mapping of RIDM and IPPD processes to generic reliability and safety based approach

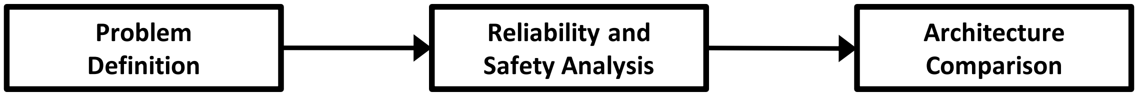


Figure 8: Generic reliability and safety based decision-making process

3.2 Research Question 1: Desired Output for Architecture Comparisons

After developing the generic process for reliability and safety based decision-making, the CONTRAST method can be derived. As stated in the previous section, the reliability and safety analyses performed within the second generic step will be dependent upon the desired output of the method. Therefore, the logical starting point for development is to look at the final step, Architecture Comparison. With the research objective and associated requirements in mind a desired output format can be selected, which will enable the successful comparison of vehicle architectures based upon their effects on reliability and safety. To address the output format, research question 1 was posed.

Research Question 1

What form of reliability and safety information is necessary to enable comparisons between unique launch vehicle concepts during early design?

In order to answer research question 1, a review of the output formats of existing reliability and safety techniques can be used. The following section will therefore review existing reliability and safety techniques in order to identify the options for output format. After identifying the output options, the derived requirements for research objective completion will then be applied to help select the desired format. This discussion is presented in Section 3.2.2, which results in the statement of an assertion to research question 1.

3.2.1 Existing Reliability and Safety Methods

In Section 2.2 current reliability and safety programs for launch vehicles were discussed. These programs included relevant handbooks, guidelines, and requirements documents from NASA, DoD, and ECSS. From these documents a list of common techniques for reliability and safety assessment during early design can be identified. The following sections review each of these techniques in more detail in order to identify the various output formats. The methods reviewed in this section represent both qualitative and quantitative approaches and are presented in no particular order.

3.2.1.1 Stress-Strength Analysis

Stress-Strength interference theory is a general model of reliability that calculates the probability that the stress applied to a component does not exceed its strength [46, 106]. Figure 9 illustrates a notional Stress-Strength diagram, which shows the probability density functions of the applied stress and component strength. The area labeled interference is the area where a component failure may occur. The reliability can be written as the probability of the stress, s, being less than the strength, S:

$$R_c = P(s < S) \tag{1}$$

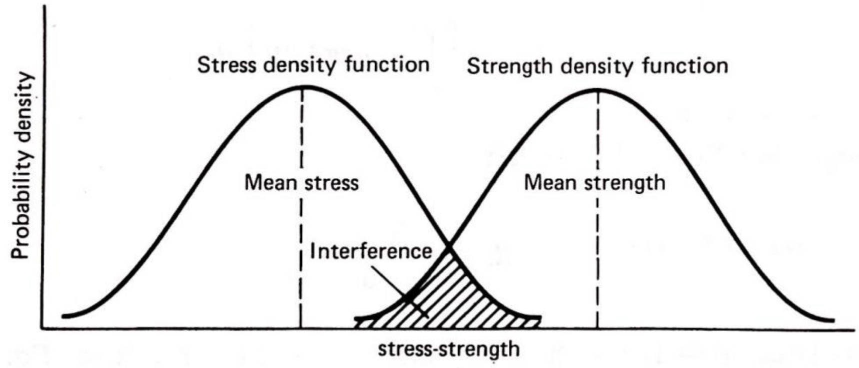


Figure 9: Notional Stress-Strength Diagram [46]

In order to calculate the value of the component reliability from the Stress-Strength diagram, the probability density functions of the curves must be known. Based upon these pdfs the reliability can be calculated using Equation 2 from reference [46]:

$$R_c = \int_{-\infty}^{\infty} f_{stress}(s) \left[\int_{s}^{\infty} f_{strength}(S) dS \right] ds \tag{2}$$

where, $f_{stress}(s)$ is the pdf of the applied stress and $f_{strength}(S)$ is the pdf of the component strength.

The Stress-Strength approach is particularly useful in situations where systems are loaded a single time. The resulting reliability estimate is the probability that the system survives the loading, thus the estimate is not a function of time [95]. Stress-Strength analysis is therefore well suited for analysis of one-shot devices such as launch vehicles [46, 95].

It is important to note that a large amount of information may be needed in order to produce a reliability estimate for a part using this approach. The stress-strength interference method assumes that the pdfs of the stress and strength are known, which may not be the case in practice. For early design these pdfs may be ill defined and require broad assumptions. This will result in a reduction in the accuracy of the reliability estimate. Depending upon the application of the Stress-Strength approach, the resulting reliability estimate will be in the form of a single point or distribution representing the expected probability of success.

3.2.1.2 Parts Count Method

The Parts Count Method (PCM) is an estimation technique that relates the number of parts found in a system to its expected reliability. Typically, the system count is defined as the total number of physically separate parts that are not composed of an assembly of smaller parts [64]. When performing the parts count, the parts are usually divided into generic type classes [42]. The method then assumes a constant failure rate for each of the type classes, which can be used to estimate the reliability of the system [42]. These failure rates can be obtained from generic equipment failure rate databases such as MIL-HDBK-217F, which contains failure data for electronic equipment [44].

Due to its simplicity, the parts count method is very useful for generating comparisons and approximate reliabilities for systems with different configurations during preliminary design [4]. It is important to note that this approach is most useful during the later portions of the preliminary design phase when the number of parts in each generic category is not expected to change [42]. The PCM approach does not, however, require very detailed knowledge of the layout of the parts within the system as it assumes that all parts are in series [42]. The use of generic failure rate data also allows PCM to produce reliability estimates without detailed design information such as part stress levels [138].

The widely accepted approach for performing PCM also includes a factor for the quality of a given part. This quality factor can be used for part types where quality level data is available [42]. The quality factor is included in the model in order to capture differences between the part types such as manufacturing tolerance. This factor can also account for maturity of the manufacturing process [44]. A quality factor of 1 can be assumed for parts that are procured in accordance with applicable specifications [42].

Equation 3 below gives the generic form of the PCM equation for the total system failure rate [42, 44]. In this equation λ_{System} is the overall failure rate for the system, n represents the number of different generic part types, N_i is the quantity of the i^{th} generic part, λ_{G_i} is the generic failure rate for the i^{th} generic part, and π_{Q_i} is the quality factor for the i^{th} generic part.

$$\lambda_{System} = \sum_{i=1}^{n} N_i(\lambda_G \pi_Q)_i \tag{3}$$

The PCM approach is very relevant to the problem being addressed within this thesis due to its simplicity. The parts count method could easily be applied to estimate launch vehicle reliability during early design. Although the PCM approach is applicable during early design and is particularly useful for system architecture comparisons, the absolute values of the estimates are not very precise [4].

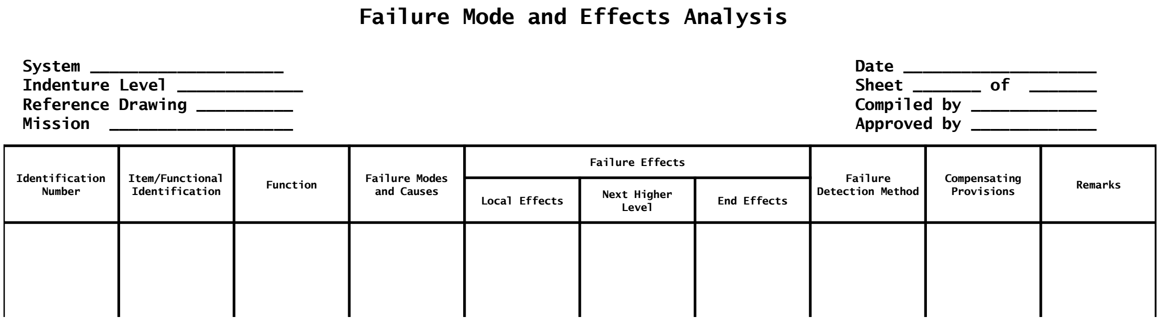
Similar to the Stress-Strength method, the output format of PCM is dependent upon the approach for generating the model assumptions. If distributions are used for the input variables the output will be a distribution of the expected system reliability. Alternatively, if single numbers are used for each input the PCM output will be a point estimate of the system reliability.

3.2.1.3 Failure Mode and Effect Analysis

Failure Mode and Effect Analysis (FMEA) is a proactive tool for discovering and correcting design deficiencies [176]. It utilizes subject matter expert input and historical data to identify potential failure modes in a system and to evaluate the potential consequences of each. A properly executed FMEA can help to [152]:

- Identify known and potential failure modes
- Identify causes and effects of each failure mode
- Prioritize the identified failure modes according to the risk priority, the product of frequency of occurrence, severity, and detection
- Provide for problem follow-up and corrective action

An FMEA is typically carried out by a team of engineers that possess the required knowledge of the system being analyzed. This team will produce the FMEA worksheet that contains all information that is generated for the identified failure modes. Figure 10 shows an example FMEA worksheet recreated from reference [40].



Failure Mode and Effects Analysis

System ____________________
Indenture Level ____________
Reference Drawing __________
Mission ___________________

Date ____________________
Sheet _______ of _______
Compiled by ____________
Approved by ____________

Identification Number	Item/Functional Identification	Function	Failure Modes and Causes	Failure Effects			Failure Detection Method	Compensating Provisions	Remarks
				Local Effects	Next Higher Level	End Effects			

Figure 10: Sample FMEA Worksheet

As can be seen in the figure, the FMEA worksheet contains columns for system function, failure mode, effects of the mode, causes, detection methods, and recommended preventative actions. The FMEA worksheet is typically populated using a well defined process [40, 172, 176]. The general process is as follows:

1. Define the system to be analyzed
2. Construct a physical or functional block diagram

3. Identify all potential item and interface failure modes and their effect on the immediate function or item

4. Evaluate the severity of each failure mode

5. Identify failure mechanisms for each failure mode

6. Identify potential corrective design or actions required to eliminate the failure or control the risk

7. Document actions taken, improvements, and notes

There are four primary types of FMEA; system, design, process and service [152, 176]. The system FMEA is used to evaluate systems and subsystems during the design phases. It focuses on potential failure modes between the functions of the system caused by system deficiencies, and includes interactions between the defined system elements [152]. The goal of a system FMEA is to rank the list of failure modes and identify mitigation strategies for each. It is primarily used to analyze and prevent failures related to technology and system configuration [176]. Since the system FMEA is typically carried out during very early design, it is sometimes referred to as a concept FMEA [176].

The second type of FMEA is the design FMEA. A design FMEA is used to analyze products before they are released to manufacturing [152]. This FMEA is typically performed as soon as the first version of the design is available [176]. Its primary purpose is to eliminate or alleviate any critical failure modes that exist within the design. The design FMEA is thus used to develop production control plans, verification, and service strategy. The output from the design FMEA serves as an input to the following process FMEA.

A process FMEA is used to identify failure modes and assess the risk of failure of the manufacturing and assembly processes. The process FMEA is typically carried out in a series of steps to include labor, machine, method, material, measurement, and environment considerations [152]. The main purpose of this FMEA is to identify ways in which the process could fail to meet the requirements and/or design intent [176]. The output of this FMEA helps to ensure a manufacturing process that is under tight control and contains a limited number of failure modes.

The fourth type of FMEA is the service FMEA. A service FMEA is used to analyze services before they reach the customer [152]. This FMEA focuses on failure modes such as tasks, errors, or mistakes that are caused by process deficiencies. The service FMEA can aid in identifying monitoring strategies, potential errors, potential bottlenecks, and critical tasks. Thus the service FMEA helps to analyze work flow by identifying critical paths.

All four types of FMEA share common benefits. The benefits of applying FMEA during the design phase of a program can ultimately help to improve the quality, reliability, and safety of a product or service [152]. First, the FMEA worksheet provides the designer with an idea of the number of failure modes that are inherent to a system. In addition, the modes are ranked based on their expected frequency of occurrence and consequences. These two pieces of information give the designer a good idea as to the reliability and safety potential of the system being considered. It also allows the designer to quickly pin-point areas of weakness in the design, which helps to prioritize any proposed fixes or proactively pursue mitigation techniques.

Although FMEA offers many benefits to a system it has major drawbacks. The foremost of these drawbacks is the time required to complete the analysis. For a simple component or element in a system an FMEA can be completed relatively quickly. However, for a complex system made up of many different components or elements the process of completing an FMEA can become very time consuming. More time is required for complex systems primarily due to the amount of detailed system information that is required. The difficulty in performing FMEA also increases as the number of possible operating modes increases, or repair and maintenance is considered [152]. Due to the time required to complete the analysis, full FMEA is not applicable for very large trade studies where multiple unique concepts are being considered. The amount of time and effort required for such a task quickly becomes unmanageable.

3.2.1.4 Preliminary Hazard Analysis

A hazard is defined as a real or potential condition that could lead to an unplanned event or series of events resulting in death, injury, occupational illness, damage to or loss of equipment or property, or damage to the environment [45]. Preliminary hazard analysis (PHA) is a technique that seeks to identify and rank hazards based upon qualitative measurement of their worst potential consequence [172]. It was originally developed by the US Army and has been proven effective in identifying hazards in the beginning of a conceptual design phase [172].

Hazard analysis is similar to FMEA in that it utilizes a worksheet as a primary guide for carrying out the analysis and storing resulting information. An FMEA worksheet will contain much of the same information as the hazard analysis worksheet, but will contain more detail. The FMEA worksheet is more detailed because it seeks to identify all potential failure modes or events that will lead to a hazardous state. In a PHA only the hazardous states will be identified, which includes hazards such as explosions, radioactive sources, pressure vessels or lines, toxic materials, high voltage, or machinery [45, 172]. For example, consider the design of a new lawn mower. A preliminary hazard analysis may identify an explosion or release of the mower blades as a hazard that could cause serious injury. In an FMEA, this hazard is identified by its underlying causes or failure modes such as; ingestion of an object causing blade failure, blade fatigue failure, or a blade failure due to material or manufacturing imperfections.

The preliminary hazard analysis process can be broken into four primary steps

[172]. The first of these steps is to define the system being analyzed. To complete this step, the overall system functionality and structure must be defined. The system structure also leads to the identification of the system boundaries between any systems with which it interacts and the domain in which it operates [172].

The second step to PHA is identification. The purpose of this step is to create a detailed list of hazards of the system. This includes the identification of events or accidents that may occur while the system is in use. The identified events or accidents are recorded in a hazards list.

The third step of PHA, assignment, begins with the hazards list. In this step, each event or accident is given a severity categorization and a predicted probability of occurrence. The assigned probability of occurrence can be developed qualitatively or quantitatively. Hazards are typically categorized as one of four classes with Class I being the most benign and Class IV the most threatening. The four classes describe hazards as negligible, marginal, critical, or catastrophic [172].

The final step of PHA is to document the findings of the analysis. During this step all the hazards and their accompanying categorizations and probabilities of occurrence are compiled into the PHA worksheet. In addition, any safety features or measures that were identified as necessary will be documented in the final report. This step ensures that the conclusions made during the application of PHA will be implemented during later phases of development.

Preliminary hazard analysis is a very powerful tool for understanding the hazards of a system. It is especially effective when applied during the early stages of design. During early design PHA allows designers to identify potential hazards in the system and begin to investigate mitigation strategies prior to full scale system development. This approach is beneficial because it helps to reduce the number of safety measures or features that need to be added to the system after development.

Preliminary hazard analysis does have limitations, however, which are due in part to its qualitative nature. The effectiveness of the PHA is dependent upon the thoroughness of the development of the hazard list. It is very possible for designers to overlook hazards during the identification phase. Using PHA also requires rigor in the documentation phase, where the results of the analysis must be properly communicated. If mitigation strategies have been identified through PHA, it is important to ensure that these strategies are actually employed. The benefits of the PHA can only be realized if the conclusions of the analysis are put to good use and the identified hazards are eliminated or mitigated.

3.2.1.5 Fault Tree Analysis

Fault tree analysis (FTA) is a failure oriented technique for calculating the reliability of a system. It provides a formal method for determination of the combinations of primary events that result in the occurrence of a specified system-level event [47]. Fault trees were conceived by H.R. Watson, after realizing that logic flow in data processing equipment could be used for analyzing the logic of system failures resulting from component failures [47]. The technique was originally applied to the Minuteman Intercontinental Ballistic Missile, which was eventually rated as one of the safest in the U.S. Air Force inventory [172].

Fault tree analysis focuses on determining the probability of occurrence of a top level undesired event. These events generally consist of complete or catastrophic failures [167]. Examples of some top level events for launch vehicles are loss of mission (LOM), loss of vehicle (LOV), or loss of crew (LOC). After identifying the top level events of interest, these events are logically branched into contributing events through cause-and-effect analysis [176]. Each of the contributing events refers to a possible cause of the top level event, which have some probability of occurrence. These contributing events, also referred to as faults, are connected to top level or lower level events via various logic “gates”. The logic gates show the relationships of events needed for the occurrence of a "higher" level event [167]. Inputs to the logic gates consist of lower level events, while the output signifies the occurrence of the higher level event [167]. When complete, the fault tree provides a graphical representation of the interaction of failures and other events in the system [172].

To illustrate the connections between contributing events, gates, and the top event a simple example is given. The example fault tree was taken from reference [167] and can be seen in Figure 11. The figure contains a top level event "T", which is connected to contributing events "A", "B", and "C". The top level event is connected directly to an "OR" gate, which has a pointed top and rounded bottom. This OR gate is connected to event C and intermediate event "A*B". Thus for the top level event to occur either event C or event A*B must occur. Event C is referred to as a basic event, which is symbolized using a circle. Basic events are the lowest level of possible failure and require no further development [167]. Intermediate event A*B, however, does require further development and is symbolized with a rectangle. In the figure, event A*B is connected to an "AND" gate, symbolized with a round top and flat bottom. This gate is connected to basic events A and B, which means that both must occur to trigger intermediate event A*B.

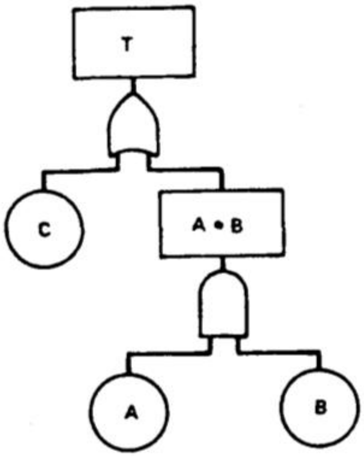


Figure 11: Simple Fault Tree

In practice fault trees will be much more complicated than what is shown in Figure 11. Many more types of logic gates and events exist for the use in complex fault trees. Other types of events include house or external events, undeveloped events, conditioning events, and transfer events [47, 167, 172, 176]. Additional logic gates include exclusive AND gates, exclusive OR gates, k-out-of-n voting gates, inhibit gates, and priority gates [47, 167, 172, 176].

It is important to note that a fault tree does not contain all possible system failures or causes for system failures. The tree is tailored specifically to its top event and thus will only include faults that contribute to this event [167]. The contributing faults that are included are also not an exhaustive set. Faults that are included in the tree are typically limited to those that have been deemed the most credible as assessed by the analyst [167].

Fault tree analysis provides many benefits to a reliability and safety program. First, FTA can yield both qualitative and quantitative information. Qualitative information from FTA includes failure paths, root causes, and weak areas of the system [176]. In this sense, the creation of the tree gives the analyst a means to review the design and to better understand functional relationships within the system. Quantitative analysis of fault trees gives a probabilistic estimate of the occurrence of the top event [176]. This probabilistic output can be used by the analyst to identify the adequacy of the design in terms of expected reliability or safety.

The results of FTA are also useful inputs to the development of verification plans, maintenance policies, and repair strategies [176]. Fault trees are applicable to many different types of systems and disciplines and can be applied at varying levels of fidelity. Thus FTA can also be used in conjunction with other techniques in order to analyze very complex systems [172].

Shortcomings of FTA are mostly seen when evaluating very large and complex systems. Creation of a full FTA for a complex system such as a launch vehicle requires a vast amount of knowledge about the individual subsystems and components as well as the operating environment. It is not uncommon for complex fault trees to contain hundreds of gates and events leading to a single top level event [171]. If the fault tree becomes very large, the process involved in quantitatively evaluating the top level event becomes very tedious. Calculations of the probability of occurrence of the top level event can become very time consuming and difficult to perform.

3.2.1.6 Reliability Block Diagrams

The reliability block diagram (RBD) is a success oriented reliability assessment technique, which is analogous to FTA. Reliability block diagrams follow the physical layout of the system using a block representation [88]. Each block represents a component in the system that has a certain probability of operating successfully. These blocks are connected by lines according to their logic relationships [176]. Depending upon the system the blocks in the diagram will be arranged in parallel, series, or a combination of the two. Other more complex arrangements are also possible, including k-out-of-n systems, cold standby, and various switches [176].

After setup, the reliability of the system is calculated using the reliability of each of the individual blocks. Based upon the set up of the diagram various paths are available for a "signal" to pass through the logic connections and blocks, typically from left to right. If a component fails, the signal is not allowed to pass through its block in the diagram. In the case that all paths are "blocked" by failed components, the system is said to have failed [95]. A simple example of a block diagram can be seen in Figure 12 from reference [42].

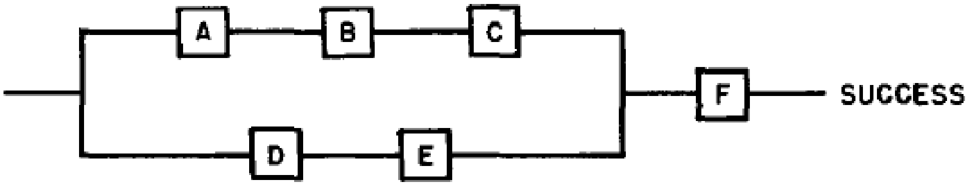


Figure 12: Simple Reliability Block Diagram

As can be seen in the figure, the system being represented has six components labeled “A”, “B”, “C”, “D”, “E”, and “F”. The right hand side of the figure is labeled success, which refers to the idea that if a “signal” can pass from the left to the right the system did not fail. In this diagram two parallel paths exist, one with A, B, and C in series and the other with D and E in series. Both of these paths then pass to component F, which in this case is critical to the success of the system. It is clear that if F fails, the path to success is completely cut. To calculate the system reliability, simple equations can be applied [176]. For blocks in series the reliability can be calculated by simply multiplying the individual component reliabilities: $R_s = \prod_i^n R_i$ [6]. For components in parallel the equation for reliability changes to: $R_s = 1 - \prod_i^n (1 - R_i)$ [6].

Although these equations are relatively simple to use, manual calculation can become very difficult as the system complexity increases. Very complex systems will often contain combinations of blocks in series, parallel, k-out-of-n, and switching configurations. For such complex diagrams, simplification methods are needed to help compute reliability. A few methods for diagram simplification include reduction method, decomposition method, and minimal cut set method [176].

3.2.1.7 Similarity Method

Similarity method is a very simplistic approach for estimating the reliability of a new system. Reliability estimates are produced via a comparison between the system under consideration and a similar system that has undergone field evaluation [42]. If the new system and the field tested system are deemed similar, then the reliability of the new system is assumed to be nearly equal to that of the old system. To complete the prediction for a new system, the data may also be subjectively adjusted upward or downward based upon the expected complexity of the new system compared to the old [109].

The similarity method allows for the designers to rapidly produce reliability estimates as well as identify differences between the new and old designs. These differences can play a large role in identifying signposts to improvements in the new design [42]. Although similarity method is rapid and simple to apply, its accuracy depends heavily upon the validity of the comparison. Prior to comparison, a set of major factors must be taken into account to ensure the validity. A set of these factors can be seen in reference [42], which are enumerated below:

1. Item physical and performance comparison
2. Design similarity
3. Manufacturing similarity
4. Similarity of the service use profile (logistic, operational, and environmental)
5. Program and project similarity
6. Proof of reliability achievement

As illustrated by the factors above, care must be taken to ensure the validity of each system comparison. If the validity of the comparison is in question, the predicted reliability of the new system must also be in question. In practice, it is difficult to ensure that each factor is met in a traceable manner. Thus, the validity of the comparison is typically left to the judgment of a subject matter expert.

3.2.1.8 Markov Chains

Markov chains were first developed based upon the work of A.A. Markov in the early 1900s [7]. Markov's study of sequences of dependent random variables ultimately lead to the creation of the term, and field, of Markov chains [7]. A Markov chain is a global state-space based representation of a system [171]. It consists of two primary elements, states and transitions. These states and transitions are conceptually simple and can be illustrated using graphical or matrix representations. A simple example of a Markov chain can be seen in Figure 13 below [171].

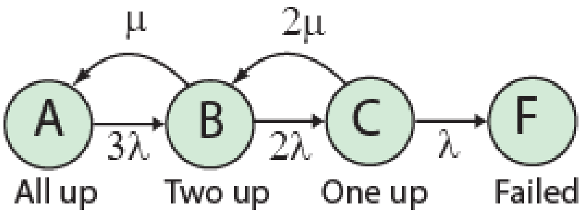


Figure 13: Simple Markov chain for triple redundant system

Figure 13 contains four states, labeled A, B, C, and F, as well as five transitions shown as arrows between the states. This example represents a triple redundant system, which has three identical components that all must fail to cause a system failure. States A, B, and C represent the system state in which 3, 2, and 1 of the components are operational, respectively. The transitions from state A to B, B to C, and C to F represent a component failure assuming failure rate λ. The upper most transitions utilize a repair rate, μ, that represents the repair of failed components.

The Markov chain operates under the simple assumption that the future state only depends upon the current state. This is sometimes referred to as the Markov property [171]. For example, in Figure 13, if the system is in state B it can transition to either state A or state C no matter if the transition to the current state was from A or C. From state B the system has a probability of transitioning to state C of 2λ and a probability of transitioning to A of μ. For this system, the Markov chain can be represented by the transition matrix Q [171]:

$$Q = \begin{pmatrix} -3\lambda & \mu & 0 & 0 \\ 3\lambda & -2\lambda - \mu & 2\mu & 0 \\ 0 & 2\lambda & -\lambda - 2\mu & 0 \\ 0 & 0 & \lambda & 0 \end{pmatrix} \tag{4}$$

The matrix Q can be read using the following conventions. Each column in the matrix represents another state of the system. In this case, column 1 represents state

A, column 2 is state B and so on. The probabilities found in each column correspond to the probability that the system will transition into the corresponding state. For example, in column 1, state A is assumed to be the current state. In row 2 of column 1 the probability of transitioning between the current state A and state B is given.

Utilizing the Q matrix, the probability of the system being in a specific state at a given step in time can be calculated. This is accomplished using Equation 5 below, where π_1 represents the vector of probabilities of the system state for time step 1 and π_0 is the initial state of the system. The initial state of the system is a simple vector with a single 1 representing the initial state with all other entries set to 0.

$$\pi_1 = Q\pi_0 \tag{5}$$

Using Equation 5 the probabilities of the system state can be calculated for any step in time using the vector of probabilities from the previous step in time. Under some conditions a limit exists, which is called a stationary distribution [171]. In this case as time becomes large the vector of probabilities will converge to some set of constant values. After this occurs, the vector of probabilities of being in the various states will be the same on all subsequent steps [73].

The primary advantage of Markov chains is the ability to model dynamic dependent events. Events such as cold spares or component repairs can be difficult to represent using RBD or FTA techniques, making it necessary to resort to Markov chains [169]. Another advantage of Markov chains is their simplicity, both conceptually and mathematically. The graphical representations of a Markov chain are very intuitive and easy to understand. Similarly, the calculations required to determine the probabilities of being in the various states can be performed very rapidly. One major caveat to these advantages is that they become less typical for very complex systems.

The reason for this caveat is that a large state-space explosion can occur for complex systems [169]. This state-space explosion is one of the primary disadvantages of the Markov chain. For very complex systems with many different components the number of states that must be represented increases greatly. The size of the model can be expected to grow exponentially with the number of modeled components [171]. Another less critical disadvantage of a Markov chain is the inability to model changes in failure rates over time. The rates defined for the original transitions are always constant, which for real complex systems may not be the case.

The two disadvantages identified above can only be avoided for complex systems by applying a different reliability technique. This technique, stochastic Petri nets (SPN), can be employed as a more compact representation of the system than a Markov chain [171]. In addition, the use of stochastic Petri nets enables the ability to change failure rates over time to more accurately represent reality.

3.2.1.9 Stochastic Petri Nets

Stochastic Petri nets (SPN) are based upon Petri net theory, which was developed from the early work of Carl Adam Petri in the 1960s [130]. Petri net theory is a graphical method that utilizes a set of basic symbols for describing relationships between conditions and events [96]. It can be used to model and analyze the dynamic behavior of complex systems [96]. Such dynamic behavior includes varying failure rates for system components or the use of cold spares [88].

Petri nets utilize a local state-space representation of the system in order to model its dynamic behavior. In graphical form a Petri net is comprised simply of places and transitions [130]. A simple example of a Petri net from reference [170] can be seen in Figure 14 below. The graphical representation of a Petri net denotes places as circles and transitions as rectangles. Directed arcs are utilized to link the various places and transitions together. Each place represents a potential state of the system being analyzed.

To identify the current state of the system, tokens are used [96]. The place in which these tokens reside represents the current state. Under certain conditions, transitions can be "fired", which based upon the directed arc connections will remove a token from a place and move it to another [169]. The firing of a transition corresponds to the occurrence of a discrete event in the system [169].

Typically, the firing of transitions is controlled using a delay. When a token enters a place with an arc connected to the input of a transition, the delay is activated. Deterministic time delays will simply fire the transition after a specified amount of time. Delays can also be represented using random variables based upon given distributions [169]. In this case the delay would have a certain probability of firing at each step in time.

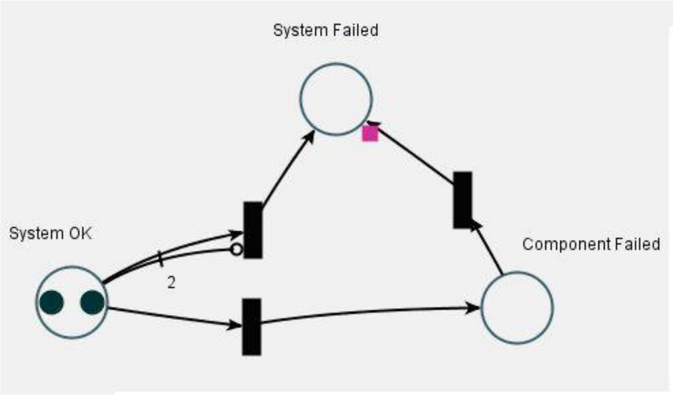


Figure 14: Simple stochastic Petri net [171]

In Figure 14 a simple SPN is given, which contains three spaces and three transitions. This example represents a system with two redundant components. If both components fail, the system is assumed to be failed. Two tokens can be seen in the space labeled "System Ok", which identifies that both of the components in the system being analyzed are currently operating. The bottom most transition links the "System Ok" space to the "Component Failed" space. This transition represents the discrete event in which one of the two components fails. Another transition is located between the "Component Failed" and "System Failed" spaces, which represents the event in which both components are failed resulting in system failure. The third transition, located between the "System Ok" and "System Failed" spaces is used to represent the change in failure rate of a single component operating by itself. After one component passes to the "Component Failed" space, this transition is activated. Upon firing this transition will pass the other non-failed component to the "System Failed" space, representing a system failure. This change in failure rate is a common dynamic effect that is easily modeled using SPN.

In order to produce reliability estimates, the SPN utilizes Monte Carlo simulation. This simulation consists of running the Petri net model many times and counting the number of times a token enters the failed state. The ratio of Monte Carlo cases in which a token entered the failed state and the total number of cases will give the probability of failure for the system.

Stochastic Petri nets are generally desirable for their ability to model systems with dynamic features [169]. Such features include dependent events and spare modeling. These events can be represented using dynamic fault trees or reliability block diagrams; however SPN offers a more compact alternative for modeling such systems. Stochastic Petri nets also benefit from continuing research, which has lead to various extensions to the SPN formulation [169]. Examples of SPN extensions include colored tokens, aging tokens, marking dependence, and trapezoidal graph representation [96, 169].

3.2.2 Assertion to Research Question 1

Section 3.2.1 presented an overview of many reliability and safety assessment techniques including FMEA, FTA, RBD, hazard analysis, parts count, similarity method, stress-strength, and reliability growth. After reviewing the existing techniques an output format for the CONTRAST method can be identified. The desired output format will provide the designer with enough information to enable differentiation between unique concepts. Another desired characteristic is the inclusion of some measure of the confidence in the resulting estimates. To begin consideration of the output formats, the methods can first be broken into two groups with FMEA and hazard analysis representing qualitative methods and FTA, RBD, similarity, parts count, stress-strength, and reliability growth representing quantitative methods.

The output of the qualitative methods, FMEA and hazard analysis, can be quickly dismissed due to their subjective rankings. Although these techniques offer strengths that will be leveraged later on, their output is not desirable for architecture comparisons. In order to evaluate many different vehicle concepts an impractical amount of time would be required to produce these qualitative assessments.

The quantitative methods have three primary types of output; point estimates, probabilistic estimates, and estimates as a function of time. The first two types of output can be produced using fault trees, reliability block diagrams, parts count, or similarity method.

Point estimates are the easiest to produce, requiring only point estimates of probability for individual blocks or events in RBD or FTA, respectively. Using RBD and FTA, only one evaluation is required to produce this type of estimate. Similarity methods also easily produce point estimates and, in the simplest form, only need to multiply each subsystem's reliability value to produce a system level value.

Probabilistic estimates can be produced via RBD, FTA, and similarity by using probability distributions for the subsystems, blocks, or events in the analysis. Utilizing these distributions, a Monte Carlo simulation can be performed, which produces the system level probability distribution. As discussed in the literature review, this distribution for a launch vehicle typically refers to the probability of loss of mission, loss of vehicle, or loss of crew. Depending upon the input distributions, the system level distribution will represent the reliability or safety of the vehicle at a specific point in time during its life-cycle.

The final type of quantitative output is the estimate as a function of time. With very special application these estimates could be produced using FTA or RBD. However, they are generally produced using reliability growth methods. Although many different techniques exist for producing growth curves the basic premise is the same. As a system matures through testing and operations, faults and deficiencies in the system will surface and be corrected. Ultimately, as these faults are corrected, they are eliminated from the system, leaving it more reliable for the next test or launch. Using this logic, each reliability growth method will produce an estimated path of reliability from the beginning of the program to vehicle maturity. For launch vehicles time will typically be represented using equivalent number of launches.

In examining the three types of quantitative outputs, the first type can be immediately eliminated from consideration. Although point estimates are the easiest to produce, the accuracy of these estimates during early design is not sufficient. Therefore the confidence in each point estimate, and ultimately the confidence in the ranking of alternatives are very low.

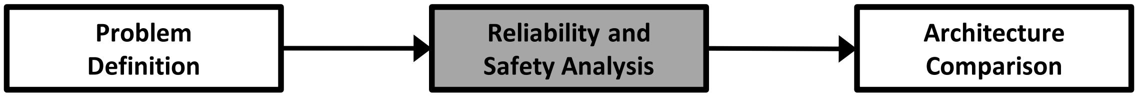
This leads to the second output type, which allows for the evaluation of the confidence in each estimate. Probability distributions allow the analyst to assess the confidence level on the estimate for vehicle reliability and safety. It also allows the analyst to evaluate the probability of reaching the reliability and safety goals for the vehicle. Although the probabilistic output is desired over the point estimates, shortcomings in this approach were identified in Section 2.3. Ultimately, the probabilistic outputs lack the ability to fully evaluate the probability that a vehicle will reach its reliability and safety goals. This shortcoming stems directly from the lack of inclusion of time in the output.

Following the previous arguments, a logical conclusion can be made that estimates as a function of time are the best suited to early design. This format gives the analyst not only an idea of the confidence in the estimates but also allows them to compare the expected reliability growth in each vehicle. Depending upon the architecture of each concept, the reliability growth curve can change in many different ways. Any changes in these curves will help the analyst make a more informed decision as to which architecture is expected to produce the highest reliability and safety. The decision will ultimately be based on more information such as; initial reliability, expected reliability at first operational flight, mature reliability, number of flights to minimum required reliability, and number of flights to maturity. From these arguments an assertion to research question 1 was produced.

Assertion to Research Question 1

Reliability estimates as a function of time are the most desirable for comparison of launch vehicle concepts during early design because they provide more information than point or probabilistic estimates.

3.3 *Research Question 2: Reliability Growth Model Type*



Now that reliability estimates as a function of time have been identified as the desired output, options for producing this output must be explored. The method development will therefore begin to address the second generic step, Reliability and Safety Analysis. In Section 3.2.1, many current reliability techniques for early design were reviewed. Of these techniques, only one is well suited for producing reliability estimates as a function of time: reliability growth methods. Since many reliability growth methods already exist they will serve as the logical starting point for determining how to produce the desired output. Research question 2 addresses the selection of a reliability growth model for application during conceptual design.

Research Question 2

What type of reliability growth model is most appropriate for producing estimates during early conceptual design?

To address research question 2, a review of existing reliability growth models must be performed. This review will determine the growth models that are most appropriate for application in early design of launch vehicles. The following section gives an overview of the three basic types of growth models. From these types a specific category is identified as appropriate for application to launch vehicle problems, which results in a detailed review of 5 different reliability growth models.

3.3.1 Reliability Growth Methods

Reliability growth methods operate with the simple assumption that as defects and faults are eliminated from a system, the system will inherently become more reliable. The idea of reliability growth can be traced back to the writings of Benjamin Gompertz, who used a learning curve type approach to evaluate human life expectancy [72]. Since that time, reliability growth has been observed in many types of complex systems including launch vehicles [61, 108].

Reliability growth methods fall into three primary categories; planning, tracking, and projection [77]. The planning and tracking categories are typically used during the primary testing phases of a program. These allow the program managers to set the reliability targets and schedule for the proposed test plan [77]. As the test program is carried out, reliability growth tracking methods can be used to measure the progress of the system. Reliability growth tracking methods give the managers an idea of how well, or poorly, the test program is progressing, which helps ensure that the system meets the reliability targets in a timely manner. The third category of reliability growth method is projection. Reliability growth projection methods are used to generate forecasts of system reliability over time. These methods allow the program managers to evaluate the expected reliability of the system and predict how long it may take for the system to reach maturity.

All three categories of growth methods will also have either a discrete or continuous formulation. Continuous growth methods track reliability in terms of total operation time, typically in hours or seconds. These methods utilized mean time between failure data to estimate the reliability of the system over time. Continuous reliability growth methods are applicable for systems in which repairs are possible, or very long duration testing can be performed.

In the discrete case, the growth models are formulated to measure time in terms of number of trials or tests. These models are applicable to "one-shot" systems such as missiles, torpedoes, or smart munitions [66, 75]. The discrete formulation of the model assumes that each trial results in a discrete success or failure event [66]. Thus for discrete growth models, time to failure is not tracked as it is in the continuous models.

A significant amount of literature exists on both the discrete and continuous side of reliability growth planning, tracking, and projection. In Table 1 a list of growth models for each different type is given. The models have been sub-divided into the planning, tracking, and projection types and then split into either continuous or discrete. It is important to note, however, that not all of the models listed in Table 1 are applicable to the launch vehicle problem being considered in this thesis. Due to the fact that launch vehicles are considered as "one-shot" systems, a discrete growth model is most appropriate. In addition, this thesis is addressing the prediction of launch vehicle reliability during early conceptual design, which corresponds to the projection type growth models. For this reason, the reliability growth methods reviewed in this section will only include the discrete projection type models. As seen in Table 1 these models include AMSAA-Crow, Hall, and Morse. Two other discrete models will be added to the review, Fries and Finkelstein, which can be seen in the discrete planning and tracking lists. The primary use of these models is for planning and tracking reliability growth, however, with special application they can be applied for projection.

Table 1: Existing Reliability Growth Models

Model Type	**Continuous**	**Discrete**
Planning	AMSAA-PM2 [164] Duane [48] MIL-HDBK-189 [41] Selby-Miller [149]	AMSAA-PM2 [164] Finkelstein [57] Fries [66]
Tracking	AMSAA-RGTMC [164] Duane [48]	Finkelstein [57] Fries [66]
Projection	AMSAA-Crow [35] AMPM-Stein [51] Corcoran [31] Crow Extended [37] Ellner-Wald AMPM [52]	AMSAA-Crow [36] Hall [75, 76] Morse [108]

3.3.1.1 AMSAA-Crow Growth Model

The derivation of the AMSAA-Crow model can be traced back to the learning-curve type approach taken by Duane in his well-known continuous reliability growth model [65]. The Duane model is based upon an observed relationship between empirical failure data for a variety of different complex systems [65]. By plotting failure data on a log-log scale over cumulative test time, Duane observed that the cumulative failure rate decreased nearly linearly [48]. From this observation, Duane introduced an approximate functional relationship between the expected cumulative number of observed failures and the cumulative test time [48, 66]. This relationship can be written, $E\{K(T)\} = \lambda T^{\beta}$, where β and λ are scale and shape parameters, T is the cumulative test time, and K(T) denotes the cumulative number of observed failures [66].

To derive a discrete version of this model a reformulation can be used in which the cumulative test time is simply replaced by the trial number. Using this reformulation, Crow derived the AMSAA-Crow model:

$$R_i = 1 - \lambda[(T_i)^{\beta} - (T_{i-1})^{\beta}]/N_i \tag{6}$$

Where, T is the trial number, N is the number of trials per test configuration, λ represents the reliability of the initial configuration, and $(1-\beta)$ is the growth parameter. The configuration number, i, is assumed to change whenever design changes have occurred. These design changes are made in direct response to observed failures.

There are two primary assumptions used during the derivation of the AMSAA-Crow model [66]. These assumptions are enumerated below.

1. The number of trials per test configuration is fixed in advance
2. The number of failures per test configuration is unknown before testing on that configuration is initiated

The first assumption implies that the number of trials does not depend on the test outcomes. In this case, testing will not be halted if a failure is experienced. Due to the fact that testing is not halted after a single failure, the second assumption follows logically as it is unknown how many times the system will fail throughout testing. Together these assumptions correspond to a case in which a batch of systems is delivered for testing, all of the systems are tested, and the results are then used to make design changes. If design changes are made a new batch of systems, representing a new configuration number, would be tested. Note that with the AMSAA-Crow formulation, the next batch of test articles does not need to be of the same number as the previous. An adjustment of this assumption was used by Finkelstein to derive a similar discrete growth model [66].

3.3.1.2 Finkelstein Growth Model

The Finkelstein growth model can be described as a special case of the AMSAA-Crow model in which the number of trials for each test configuration is constant [66]. In the case where $N_i = N$, the reliability growth model can be reduced to:

$$R_i = 1 - \lambda N^{\beta-1}[(i)^{\beta} - (i-1)^{\beta}] \tag{7}$$

Where i is the configuration number, N is the number of trials per test configuration, λ represents the reliability of the initial configuration, and $(1-\beta)$ is the growth parameter. As with the AMSAA-Crow model, the configuration is assumed to change when design fixes are implemented to eliminate specific observed failures.

The two primary assumptions for the AMSAA-Crow model also hold for the Finkelstein model, with one minor adjustment. The second assumption remains unchanged, which refers to the number of failures per test configuration being unknown before testing is initiated. The first assumption varies slightly due to the special case that was used to derive the Finkelstein model. The first assumption states that the number of trials per test for each configuration is fixed. For the AMSAA-Crow model this number is fixed for each configuration but can vary between configurations. The Finkelstein model, however, assumes that this number is constant for all configurations.

Due to this assumption, the Finkelstein model is a special case of the AMSAA-Crow model, $N_i = N$. This conclusion means that the two models are nearly identical with their only variation being in the assumed test strategy. The AMSAA-Crow test strategy allows for changes in the number of tests after a configuration change, while the Finkelstein model assumes no change in number of tests. In order to more accurately reflect the test strategy for expensive one-shot systems, Fries offers an alternative derivation in reference [66].

3.3.1.3 Fries Growth Model

The Fries growth model was derived as a discrete learning-curve based reliability growth model [66]. As discussed above this model is very similar to the AMSAA-Crow and Finkelstein models. The primary difference lies in the assumed test strategy. Fries derived primary assumptions based upon a test strategy for destructive testing of very expensive one-shot systems [66]. In this case, testing is halted after any failure has occurred and design fixes are implemented before testing continues. With this assumption, the results for each test configuration consist of a string of successes, possibly zero, followed by a single failure [66]. This single failure then results in a change in design configuration. The adjustments made by Fries to the assumed test strategy results in two primary assumptions for the model, which are enumerated below.

1. The number of trials per test configuration is unknown before testing on that configuration is initiated
2. The number of failures per test configuration is fixed at one

The first assumption stems from the test results consisting of a run of successes prior to a failure. The trial at which the failure occurs is unknown, therefore the total number of trials for the configuration is unknown. After a single failure is experienced the testing is halted and a design correction is sought. This leads to the second assumption, which makes intuitive sense for destructive testing of expensive systems. After a single failure, testing will be halted in order to avoid excessive cost commitment to testing a configuration that will be changed at the end of the test phase anyways.

Using these two assumptions, Fries derived a new expression for system reliability:

$$R_i = 1 - \lambda[(i)^\beta - (i-1)^\beta]^{-1} \tag{8}$$

Where i is the configuration number and β and λ are parameters of the learning curve property. As Equation 7 and Equation 8 illustrate, the Finkelstein and Fries models are very similar with the primary difference being the removal of the number of trials from the expression for reliability.

3.3.1.4 Morse Growth Model

The Morse reliability growth model was developed as a practical method for generating growth forecasts for new systems in the space industry [108]. The intent of the model is to clearly identify and quantify the primary drivers of reliability to yield a model that is mathematically sound and directly amenable to systems engineering inputs [108]. The Morse model was derived as a discrete reliability growth model, which projects vehicle reliability versus number of flights. It also utilizes three fundamental assumptions in regard to defect handling and system configuration. The three primary assumptions used for deriving the Morse model are enumerated below.

1. Any mission failure, or any anomaly recognized as having a significant potential to cause mission failure, will result in the application of all practical attempts at eliminating the source of failure before the next flight

2. A new system is manufactured for every flight; there is no wear-out effect with increasing flight number

3. The system design and processes (manufacturing, operations) are changed only for the purpose of increasing reliability

Using these three primary assumptions, Morse represents the primary drivers of reliability growth as a set of probabilities. These probabilities include; the probability of defect trigger, conditional probability of defect detection if triggered, conditional probability of proper characterization of the defect given detection, and the conditional probability that the defect is eliminated from the system given it was detected and characterized. In Figure 15, a diagram of the Morse model is shown.

This diagram illustrates the linkage between each of the model parameters, which are the probabilities noted in green. The diagram also includes a conditional probability of failure isolation given a defect trigger, as well as a conditional probability of crew escape given a loss of mission event. The addition of the crew escape probability is only necessary for launch vehicles in manned configuration.

From Figure 15 a list of parameters for the Morse model can be created. These parameters, shown in Table 2, are used to develop a mathematical expression for expected vehicle reliability at flight N. In all, Morse utilizes seven different parameters in the growth model. It is important to note that Morse's formulation also allows for the inclusion of different defect types, which gives the ability to model severe defects and minor defects individually. In the case where multiple defect types are included, the defect type is represented by the index k.

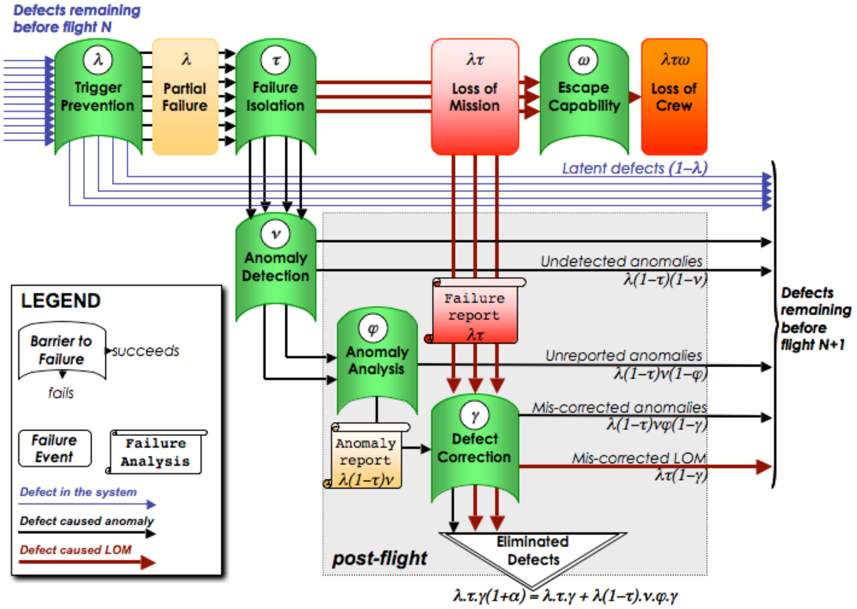


Figure 15: Diagram of Morse Reliability Growth Method [108]

Table 2: Morse reliability growth model parameters

Parameter	**Definition**
d_k	Initial number of defects of type k
p_{min}	Minimum probability of failure
λ_k	Probability of defect trigger
τ_k	Given defect trigger, conditional probability of LOV
ν_k	Given anomaly, probability that it is observable
ϕ_k	Given observable, probability that anomaly is noticed and reported
γ_k	Given reported, probability that defect is eliminated

The derivation of the Morse model, utilizing the parameters from Table 2 is as follows. First, the probability that a defect is still present at flight N is calculated:

$$\delta_k(N) = [1 - p_k(\gamma_k + \alpha_k)]^{N-1} \tag{9}$$

Where p_k is the probability of system failure for the given defect and α_k is a measure of the potential to eliminate a defect before it causes loss of vehicle:

$$p_k = \lambda_k \tau_k \tag{10}$$

$$\alpha_k = (\eta_k - 1)\rho_k \tag{11}$$

The parameters η_k and ρ_k are the mean number of defect triggers to cause LOM and the conditional probability to eliminate a defect after causing a partial anomaly, respectively.

$$\eta_k = 1/\tau_k \tag{12}$$

$$\rho_k = \nu_k \phi_k \gamma_k \tag{13}$$

With all the parameters now known, the reliability of the entire system at cycle N is:

$$R(N) = (1 - p_{min}) \prod_{k=i}^{D} [1 - \delta_k(N) p_k]^{d_k} \tag{14}$$

Using Equation 14, the Morse model can be applied to project the reliability of a launch vehicle. The projection requires assumptions for the seven primary parameters from Table 2 as well as an assumed number of trials, flights, cycles, etc.

3.3.1.5 Hall Growth Model

The Hall reliability growth model is a discrete model for one-shot systems, which is derived in references [76] and [77]. Hall indicated that the area of discrete reliability growth projection was underdeveloped, which motivated the derivation of a new model [76]. The derivation of the model follows five primary assumptions, which are enumerated below.

1. A trial results in a dichotomous success/failure outcome such that $N_{i,j} \sim Bernoulli(p_i)$ for each failure mode $i = 1, ..., k$ and trial $j = 1, ..., T$
2. The distribution of the number of failures in T trials for each failure mode is binomial
3. Initial failure mode probabilities of occurrence $p_1, ..., p_k$ constitute a realization of an s-random sample $P_i, ..., P_k$ such that $P_i \sim Beta(n, x)$ for each $i = 1, ..., k$
4. Potential failure modes occur s-independently of one another and their occurrence is considered to constitute a failure
5. There is at least one repeat failure mode

In examining the Hall model assumptions, a set of four primary parameters can be identified. The first of these parameters stems from assumptions 1 and 3, which is the number of failure modes, k. This parameter represents the inherent number of failure modes that exist in the system. The second parameter stems from assumption 1, which is the number of trials during the test phase, T. For application to a launch vehicle, this parameter would represent the equivalent number of flights. Finally, assumption 3 states that the initial probabilities of failure for each failure mode are random samples from a Beta distribution. This implies that the two scale and shape parameters of the distribution are also primary model parameters.

Using the four identified parameters Hall first evaluates an indicator function, which is nothing more than a history of occurrences of all the failure modes. The indicator function can be evaluated by treating each step in time as an independent Bernoulli trial with probability of failure equal to the probability of occurrence for each failure mode. The indicator function, $I_i(t)$ can be written:

$$I_i(t) = \begin{cases} 1 & \text{if failure mode i is observed on or before trial t} \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

From the indicator function the model of reliability at trial t can be logically derived. The resulting reliability estimate at trial t can be written:

$$r(t|\vec{p}) \equiv \prod_{i=1}^{k}(1 - [1 - I_i(t-1) * d_i] * p_i) \tag{16}$$

Where I_i is the indicator function from Equation 15, p_i is the probability of occurrence for failure mode i, and d_i is the fix effectiveness factor (FEF) for failure mode i.

Equation 16 introduces a new variable into the Hall model, the FEF. The FEF is a representation of the expected percent reduction in the probability of occurrence of a failure mode after corrective action has been taken. An FEF of 1 implies that a failure mode will be completely eliminated from the system if encountered. On the opposite end of the spectrum, an FEF of 0 implies that no corrective action will be taken if the failure mode is experienced.

To evaluate the FEF for each failure mode Hall utilizes another Beta random variable. This requires the addition of two parameters, which are the scale and shape parameters of the FEF Beta distribution. The resulting Hall model can be applied using Equation 16 along with assumptions for the number of failure modes, scale and shape parameters for the distribution of probabilities of occurrence, and the scale and shape parameters for the FEFs.

3.3.2 Hypothesis 2

The previous section presented a detailed description of five growth methods that were deemed applicable during conceptual design of launch vehicles. This discussion illustrates that all of the models utilize different assumptions in order to capture the growth trends. Therefore to form a hypothesis to research question 2, the generation of these assumptions is a primary consideration.

The first three models discussed in Section 3.3.1, AMSAA-Crow, Finkelstein, and Fries are very similar in their assumptions. All of these models utilize a growth parameter, which is derived using reliability data from a "surrogate" system. The AMSAA-Crow and Finkelstein models have additional parameters relating to the test program that the vehicle will go through during development. These assumptions include the number of test configurations and the number of trials per configuration.

The last two models, Hall and Morse, utilize different assumptions regarding the number of failure modes inherent to the system. The Hall model includes parameters for the number of failure modes, probability of occurrence of the modes, and fix effectiveness. The Morse model also includes a parameter for the number of failure modes, but takes a different approach than Hall. This approach requires parameters for probabilities of occurrence, detection, action, and correction.

In order to identify the most appropriate reliability growth method two motivating questions can be posed. The first question asks, what vehicle information is available during conceptual design? The design information is an important consideration because it will have a large impact on the traceability and accuracy of the resulting reliability estimate. Therefore, it will be important to select a growth method that contains assumptions that can be made during conceptual design. The available design information and its implications on the appropriate growth model are discussed in Section 3.3.2.1.

The second motivating question asks, how can a reliability estimate be produced using such little information? This question is relevant to the selection of a specific growth model because it will help identify which pieces of information are vital to reliability prediction. This information may or may not be represented in the various growth model assumptions, which will have an effect on the validity of the final results. This question will be addressed in Section 3.3.2.2.

3.3.2.1 Information Available during Conceptual Design

To begin the discussion of conceptual design information, recall the three categories of drivers of launch vehicle reliability discussed in Chapter 1. The three categories of drivers are vehicle architecture, programmatic environment, and operating environment. Each of these categories represents a set of design parameters, vehicle attributes, or program details that all affect the reliability of the system. As the vehicle progresses through its life-cycle more information about each category becomes known, which allows for more accurate estimation of reliability.

The first category, vehicle architecture, refers to the physical description of the system. This category includes parameters such as number of engines, number of stages, engine cycle type, etc. The vehicle architecture category also includes any interaction effects that may occur between parts, components, assemblies, and subsystems. During conceptual design a fairly basic representation of the vehicle is available. This representation may include general descriptions of the vehicle subsystems along with some basic interactions or relationships between them.

The programmatic environment category is typically the least developed during conceptual design. This category includes items such as test planning, decision making, management style, and developer experience. During conceptual design it is very difficult to characterize the effects of management style or decision making on vehicle reliability during operations. This is primarily due to the fact that it is impossible to anticipate all of the major decision points that may arise throughout the program. In addition, the details of the development and test program will not be known. Other considerations such as the probability of defect introduction during manufacturing or integration are also nearly impossible to quantify during early design. For this reason, the programmatic environment category plays little to no role in reliability estimation during conceptual design.

The final category, operating environment, represents the operating conditions experienced by the vehicle. This category also includes the interactions between the system and the surrounding environment. During conceptual design, trajectory tools can be used to derive the basic loading conditions experienced by the vehicle. However, it is much more difficult to characterize the interactions that may occur between the vehicle and the environment. A description of the operating environment during conceptual design may only include basic information such as max dynamic pressure and max acceleration.

After considering the information available during conceptual design, two of the five identified growth models can be eliminated from consideration. The elimination of the AMSAA-Crow and Finkelstein models stems directly from the lack of knowledge in the programmatic category, which is the least developed during conceptual design. Due to the fact that the AMSAA-Crow and Finkelstein models rely upon assumptions related to test planning, it is difficult to believe that their assumptions can be made during early design. Specifically, the parameters for number of configurations and number of trials per configuration will not be known for the system and will be difficult to estimate. The only way to estimate these parameters is to compare the new vehicle to a previous vehicle, which was most likely developed under different leadership, budget, and schedule. Even if the previous vehicle was developed under the same circumstances, any changes in leadership, budget, or schedule would require adjustment to the model parameters, which cannot be anticipated during conceptual design.

It is important to note that the third model, Fries, also contains a parameter for the number of vehicle configurations. However, an argument can be made for this parameter to be substituted with the number of equivalent flights. The Fries model assumes that only one failure is experienced per test configuration, which means that the number of trials per configuration is unknown. In this case, the configuration number can simply be incremented based upon whether or not the system was assumed to fail. This can be done by taking a random draw using the current reliability value for the system. Performing such a draw eliminates the need for the assumptions that are otherwise required in the AMSAA-Crow and Finkelstein models.

3.3.2.2 Conceptual Reliability Estimation

After discussing the availability of information during conceptual design, it is important to look at how this information can be used to generate a basic reliability estimate. A simple example will be presented in this section using Stress-Strength interference theory, which will lead to observations related to the desired parameters for the selected growth method. Stress-Strength theory was presented in more detail in Section 3.2.1.1.

For the conceptual reliability estimation example, consider a simple component consisting of a bar rigidly attached to a ceiling holding a load, P. An illustration of the component can be seen in Figure 16.

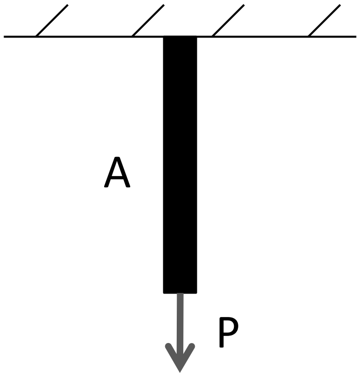


Figure 16: Bar component rigidly attached to ceiling

The only failure mode for this case is the yielding of the bar material under the applied load. This situation occurs when the applied stress is greater than the yield strength of the material. For this example, assume that both the yield strength of the material and the cross-sectional area of the bar are normally distributed. In addition, the applied load will be normally distributed. From [46] the equation for component reliability from Stress-Strength theory can be written:

$$R_c = \int_{-\infty}^{\infty} f_{stress}(s) \left[\int_{s}^{\infty} f_{strength}(S) dS \right] ds \tag{17}$$

where s is the applied stress, and the applied load, P, and the bar cross-sectional area are normally distributed:

$$s = P/A \tag{18}$$

$$P \sim N(\mu_p, \sigma_p) \tag{19}$$

$$A \sim N(\mu_a, \sigma_a) \tag{20}$$

The material strength, S, is also normally distributed with a pdf:

$$f(S) = \frac{1}{\sigma_m \sqrt{2\pi}} e^{\frac{-(S-\mu_m)^2}{2\sigma_m^2}} \tag{21}$$

In order to calculate the reliability of the component using the above equations, three parts of information are needed. The first is an estimate of the load that will be applied to the bar. This estimate may be a uniform distribution if the bar is to be used as a generic hanger, or a very tight normal distribution if the bar is intended for one load case only. The load information will define the mean (μ_p) and variance (σ_p) of the applied load, P.

Next, manufacturing tolerances for production of the bar are needed to define the cross-sectional area. The manufacturing technique used to fabricate the bar will have an effect on the tolerance of the bar dimensions. Ultimately these tolerances will define the mean (μ_a) and variance (σ_a) of the bar area, A, which will have an effect on the applied stress. The final piece of information is in regard to the bar material. Through materials testing the material yield strength can be assessed, giving the mean and variance seen in Equation 21.

It is important to note that some of this information may not be available during conceptual design of the component. For example, the manufacturing technique that will be used to fabricate the bar may not be identified early in the design process. In this example problem the lack of information is not an issue because there are very few parameters that define the stress and strength of the component. However, in the case of a large complex system, many parameters with large uncertainty will lead to either a result with high uncertainty, or the inability to produce a result altogether. This is part of the reason why producing accurate reliability estimates during conceptual design is very difficult. As this example problem will illustrate, however, it is possible to reduce the number of parameters required to produce the reliability estimate.

To complete the reliability calculation for the bar example, values were assigned to all of the stress and strength parameters. These parameters can be found in Table 3 below. Using the parameters from the table the applied stress distribution can be approximated numerically. Figure 17 below shows the strength and stress distributions for the bar problem. The distribution on the left displays the applied stress while the right shows the yield strength.

Table 3: Stress-Strength Parameters for a simple bar component

Parameter	**Value**
μ_p	4,250 lbs
σ_p	250 lbs
μ_a	0.5 in
σ_a	0.0025 in
μ_m	10,000 psi
σ_m	10 psi

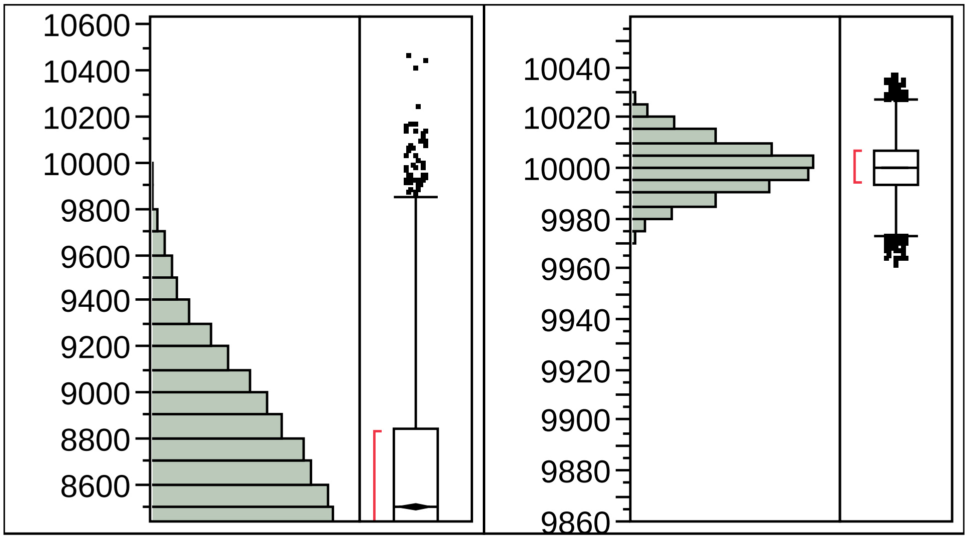


Figure 17: Distributions for applied stress (left) and yield strength (right)

Now that the stress and strength distributions have been defined, the reliability of the component can be calculated, which yields $R_c \approx 0.9989$. Since the reliability of the component is now known and the component only has a single failure mode, the probability of occurrence of that mode can be written as $P_f \approx 0.0011$.

Next, consider the case where an additional failure mode is introduced into the component. In this case the bar is now attached to the ceiling using an adhesive with the yield strength, S_A. This yield strength will be assumed to be a function of the ambient temperature, T. It can also be assumed that the adhesive failure mode and the bar material failure mode are independent. This assumption is valid because one failure mode does not cause the other and a case where both modes occur simultaneously is not expected.

In order to calculate the reliability of the component in this situation the original six parameters are needed along with three new ones for the second failure mode. The three new parameters are the mean (μ_T) and variance (σ_T) of the ambient temperature and the relationship between the adhesive yield strength and temperature.

The ambient temperature parameters can come from local weather history if the final operating location of the component is known. If this location will be random or is unknown, then a fairly large spread of temperatures will need to be considered. For this example, let us assume the component is used in a climate where the temperature is relatively stable with a mean of 80 degrees and a variance of 5 degrees.

The relationship between the adhesive strength and ambient temperature is more difficult to derive. If this component were being developed the relationship could most likely be derived by directly testing the yield strength of the adhesive at various temperatures. For the example problem, a simple quadratic relationship will be assumed, which can be written as: $S_A = S_{max} - \frac{1}{4}T^2$, where S_{max} is the maximum yield strength of the adhesive.

Using the new parameters the reliability for the adhesive can be solved for numerically. The resulting reliability of the adhesive is $R_{ad} \approx 0.9982$, with the probability of occurrence of this mode being $P_{f_{ad}} \approx 0.0018$. After calculating the reliability for the new mode, the overall component reliability can be calculated as $R_c * R_{ad} \approx 0.9971$.

At this point, an important observation can be noted. To generate the reliability value for the component assumptions, nine different parameters were needed. However, the reliability could have been calculated in alternative fashion using the number of failure modes multiplied by the probability of failure mode occurrence:

$$R_c = \prod_{i=1}^{N}(1 - P_{f_i}) \tag{22}$$

where N is the number of failure modes and P_{f_i} is the probability of occurrence of failure mode i. This alternative approach would produce a reliability estimate for the component that requires only two parameters instead of nine. To produce a reliability estimate for the component using Equation 22 all that is needed is an estimate for the probability of occurrence of the two failure modes. The second parameter, number of failure modes, is easy to quantify for the example problem because there are only two modes. Therefore, there is no uncertainty in this parameter. The equation for the reliability of the bar with two failure modes becomes:

$$R_c = (1 - P_{f_1}) * (1 - P_{f_2}) \tag{23}$$

where P_{f_1} is the probability of occurrence of the bar material failure and P_{f_2} is the probability of failure for the adhesive. Without knowing details in regard to the operating environment, material properties, or adhesive strength vs. temperature relationship the component reliability can now be calculated by applying distributions for the probabilities of occurrence.

The reduction in the parameters required for reliability estimation is especially relevant to the launch vehicle problem being addressed in this dissertation. As was discussed in Section 3.3.2.1 there is very little information available during conceptual design. This lack of information may make it difficult to estimate parameters such as manufacturing tolerance or ambient temperature. Therefore, it would be desirable to select a growth model for application to the launch vehicle problem that shares similar assumptions.

The simple bar problem illustrates how the detailed reliability analysis can be simplified into a number of failure modes approach, which is actually reflected in a few of the growth models. The Hall and Morse models specifically, utilize the same approach as in Equation 22 with one or more additional parameters. After carrying out the bar example these growth models become much more attractive especially in comparison to the third model under consideration, Fries.

The bar problem traces the number of failure modes and probability of occurrence assumptions back to a more detailed reliability analysis. However, parameters such as the growth parameter found in the Fries model are much more ambiguous in nature. This growth parameter cannot be traced back to the more detailed reliability analysis. For this reason, the Fries model can be eliminated from consideration. The Hall and Morse models are now the primary options for application in the CONTRAST method.

3.3.2.3 Growth Model Selection

After narrowing down the reliability growth options to the Hall and Morse models, one must be selected for application in the CONTRAST method. Throughout the discussion of design information and conceptual reliability estimation in Section 3.3.2.1 and Section 3.3.2.2 two criteria for selecting a model became apparent. These criteria are the accuracy of the model estimates and the traceability of the model parameters. Ultimately, the traceability of the model parameters is the most important criteria for selection because its inherent effect on the accuracy of the output. In addition, traceable assumptions will allow for more transparency during the reliability assessment as well as more confidence in the final result. To develop hypothesis 2, the traceability of the two remaining growth models was considered.

The first growth model option is the Hall model, which was discussed in detail in Section 3.3.1.5. The Hall model utilizes a number of failure modes approach similar to what was shown in Section 3.3.2.2 for the bar example problem. The Hall model introduces one more parameter, called the fix effectiveness factor, in addition to the number of failure modes and the probability of occurrence. The fix effectiveness factor represents the percent reduction in the probability of occurrence of a failure mode given that the mode occurred and a fix was implemented.

All of the parameters within the Hall model can be estimated in a traceable manner. The number of failure modes inherent to a system can be estimated using techniques such as FMEA or PCM, while the probabilities of occurrence can be estimated using historical failure data. The fix effectiveness factors are the least traceable of the Hall model parameters. This is due to the fact that the amount of reduction of the probability of occurrence may not be directly calculable from historical data. However, this parameter can still be estimated via a review of documents pertaining to failure reporting and correction from previous programs. Due to the fact that the Hall model relies on three parameters that can be derived in a traceable manner, it is considered to be a more desirable option than the Morse model.

The second option, the Morse model, was discussed in detail in Section 3.3.1.4. The Morse model utilizes various probabilities associated with failure mode trigger, observation, reporting, and correction. Two of these assumptions, the number of failure modes and the probability of failure mode trigger are directly in line with the bar example problem from Section 3.3.2.2. The addition of other parameters, however, results in a rather large number of assumptions that must be input into the model. In all there are 7 parameters of the Morse model that require assumptions. These parameters are tabulated in Table 2 in Section 3.3.1.4.

The high number of assumptions required by the Morse model is detrimental when considering traceability. Although the parameters used in the Morse model make logical sense and play a large part in the reliability growth of real systems, they are very difficult to estimate. For example, one of the model parameters refers to the probability that an anomaly is detected during the flight given that an anomaly has occurred. Obviously for corrective actions to be taken and reliability growth to occur the anomaly must be detected during the flight. However determining the probability that an anomaly is detected requires knowledge of the number of anomalies that were not detected. This information is truly unknown, and it is very apparent that having an accurate estimate of this number is non-sensical. Due to the inability to accurately assess the Morse model assumptions it is considered to be the worse than the Hall model in terms of traceability of assumptions.

Hypothesis 2 states that the Hall model will be more desirable for application in the proposed approach based upon a consideration of the traceability of the model assumptions. This hypothesis will be tested by Experiment 1 in the following section.

Hypothesis 2

If the Hall growth model is applied, the output reliability estimates will have greater accuracy while utilizing more traceable assumptions than the AMSAA-Crow, Finkelstein, Fries, or Morse growth models.

3.4 Experiment 1

After developing hypothesis 2 in the previous section a test is needed to substantiate the hypothesis. In order to test the hypothesis, Experiment 1 will address two primary considerations for the growth model. First, the traceability of the model parameters is of utmost importance to the method. Utilizing a model with traceable parameters will provide the designer with a more defensible reliability output. Additionally, traceable parameters will allow for a more accurate reliability prediction, which is another primary consideration in selecting a growth model. The goal of Experiment 1 is therefore to further assess the assumption traceability as well as test the model prediction accuracies.

3.4.1 Experimental Setup

Experiment 1 will test the Hall and Morse models by considering the traceability of their assumptions and the accuracy of their reliability estimates. In order to consider traceability, a qualitative assessment will be presented. This assessment will be based on the detailed descriptions of the models given in Section 3.3.1.4 and Section 3.3.1.5. To assess the accuracy of the reliability estimates a quantitative experiment can be carried out. In order to make a quantitative assessment of the model accuracy, reliability data from previous vehicles will be required. Reliability predictions for these previous vehicles will be produced using each model and the results will be compared to the historical data.

3.4.1.1 Traceability of Model Assumptions

The traceability of the Morse and Hall models was discussed very briefly in Section 3.3.2.3. In this section all of the parameters for both models will be discussed in more detail. After assessing these parameters conclusions will be made as to which growth model can be considered as more traceable.

The first model to be discussed is the Morse model. In Section 3.3.1.4 seven model assumptions were identified, which were tabulated in Table 2. The first parameter to discuss is the initial number of defects, d_k. This parameter is equivalent to the number of failure modes assumption found in the Hall model. As discussed in Section 3.9 this assumption can be produced using multiple different approaches. If the vehicle being considered is similar to a previous vehicle, failure data or detailed design information from the previous vehicle can be used to generate this assumption. Alternatively, a parts count approach or failure mode and effect analysis can be used to produce a value for d_k. Considering the multitude of origins for this parameter it is considered one of the more traceable assumptions contained within the Morse model.

The second assumption for the Morse model is the minimum probability of failure of the system, p_{min}. This assumption is equivalent to the maximum expected reliability of the system being analyzed. If the Morse model is run over a very long number of time steps the reliability of the system will approach $(1 - p_{min})$. Due to the fact that this assumption sets the upper bound for the reliability output of the model it is important to set the appropriate value. In order to generate this assumption in a traceable manner we must have some sort of previous system or experience to draw upon.

To generate the assumption using a previous system, the appropriate data must exist for estimating the previous system's mature reliability. It is important that the mature reliability is known because it will most accurately represent the minimum probability of failure of the system. A similar approach can be taken in using previous experience to estimate this parameter. Using this approach will probably require an order of magnitude type estimate, which says at best this system will have a probability of failure of 1 in 100 or 1 in 1000. In reality this assessment will be based upon the perceived complexity of the system with a more complex system being given a higher minimum probability of failure. The second Morse model assumption can be considered traceable if it is derived from a previous system. However, if previous experience is used, caution must be taken in order to avoid biasing the reliability estimate towards the over cautious or over optimistic.

The third Morse model assumption is the probability of defect trigger for each of the defects identified by the first assumption. This assumption is the same as the probability of occurrence parameter found in the Hall model. In order to accurately estimate this parameter previous data must be used. For components or parts, many different sources exist that contain failure rate data. Examples of such sources include the Military Handbook for Reliability Prediction of Electronic Equipment [44], or the Reliability Analysis Center Failure Mode/Mechanism Distributions [23]. Both of these sources contain data pertaining to many parts and components that can be found in complex systems. The data is derived from testing or operations of each part and component, which can be used to very easily generate distributions for the probability of failure of a new component.

Obviously at the component or part level these assumptions can be considered very traceable. However, at a higher subsystem level the comparison to the previous component data becomes less obvious. In order to produce subsystem or system level probabilities of failure, the component level data must be rolled up to the higher levels. This requires additional information in regard to the number of components contained within the system or subsystem, which will ultimately determine the assigned value for probability of failure.

The next assumption for the Morse model is the conditional probability of LOV given that a defect has been triggered, τ_k. This value relates to the amount of fault tolerance or redundancy within the system. If a system has a high degree of fault tolerance the occurrence of a single fault may not cause an LOV event, which equates to a low value for τ_k. In practice, this parameter is much easier to estimate if it only considers redundancy. If multiple defects are considered to be redundant, the value for τ_k can be set accordingly.

Another simplification that can be made for this parameter is to only include defects that contribute directly to an LOV event. In this case, fault tolerance and redundancy is excluded and the τ_k parameter can be defaulted to 1. For an actual launch vehicle however, fault tolerance and redundancy both come into play. During early design there is typically no information regarding fault tolerance of the system, which means that the assumption for τ_k may be difficult to trace.

The fifth assumption in the Morse model is the probability that a defect is observable given that a defect has occurred, ν_k. This parameter represents the chances that a defect will actually be detected during operations, which will give the program the opportunity to investigate the defect. An obvious source for information to derive this parameter is defect or anomaly reports from previous launches. For example, after each launch of the STS vehicle a list of anomalies was generated for future investigation [108].

The issue with the ν_k parameter is that it implies that the number of defects that have not been detected are known. Even if detailed information from previous launches is available, the number of defects that occurred that were not detected is unknown. For this reason there can never be a truly accurate assessment of ν_k because the information required to calculate it is always unknown. In order to produce this parameter a best guess must be used based upon previous experience. Due to the fact that the ν_k parameter cannot be traced back to actual data, it is considered to be very detrimental to the traceability of the Morse model.

The next assumption for the Morse model is the probability that an anomaly is noticed and reported given the anomaly was observable, ϕ_k. This parameter represents the chances that during post launch data analysis a defect is properly identified and reported for future investigation. Just because a defect may have been detected through flight data recording, does not necessarily mean that it will be investigated prior to the next flight. The ϕ_k parameter was introduced into the model to represent this situation.

In order to estimate the ϕ_k parameter data regarding post launch analyses must be acquired. This data is not necessarily cut and dried however. To estimate the probability of anomaly reporting there is a human factors effect that needs to be considered. First, the rigor of post flight data analysis will stem from the leadership and direction of the program. For example, it is expected that this analysis would be much more rigorous for a manned launch vehicle versus an unmanned vehicle. The second human factors consideration is in regard to the analyst performing the post flight data analysis. There may be instances where one analyst catches certain defects while another does not. All of these considerations make it very difficult to estimate the ϕ_k parameter, especially during conceptual design. Due to the lack of knowledge about post flight data analysis during early design, ϕ_k can only be estimated using a best guess approach. This is obviously detrimental to the traceability of the model.

The final assumption of the Morse model is the probability that the defect is eliminated from the system prior to the next launch, γ_k. This parameter is equivalent to the fix effectiveness factor that is used by the Hall model. If an anomaly was detected and reported, corrective action may be taken to reduce the probability of occurrence of the defect prior to the next flight. If the defect is completely eliminated, γ_k is equal to 1. If no corrective action is taken γ_k is equal to 0.

To estimate γ_k from previous data, reports from defect elimination efforts must be available. This includes any post launch analysis identifying the defect as well as reports from the redesign efforts. In order to estimate the parameter from this previous data, the probability of occurrence of the defect after the redesign must be known. This means that additional flight data is required in order to determine whether or not the defect occurred again after the redesign. In practice this data will be very hard to gather and interpret. The probability of occurrence of the specific defect after a redesign will be very difficult to derive from the data even if enough data exists. As discussed in Section 3.9, this assumption requires a best guess type approach.

After progressing through the Morse model assumptions, the assumptions used in the Hall model can now be considered. As mentioned above, three of the parameters from the Morse model align with the three primary assumptions of the Hall model. This is simply due to the similarity in which the models were derived using the number of failure modes and probability of occurrence approach. From Section 3.3.1.5 three parameters requiring assumptions were identified. These parameters include the number of failure modes, probability of occurrence for the modes, and the fix effectiveness factors.

The first assumption, number of failure modes, is equivalent to the number of defects assumption from the Morse model. As discussed above this type of assumption can be generated via comparison to previous systems. The number of failure modes can also be estimated using an existing technique such as parts count method or failure mode and effect analysis. Due to the availability of multiple techniques for generating this assumption it is considered to be very traceable in comparison to other model parameters.

The second assumption for the Hall model is the probability of occurrence for the failure modes. This parameter is equivalent to the probability of defect trigger in the Morse model. Thus the same discussion above for the Morse model also holds for the Hall model. The probability of occurrence can be estimated using databases of failure rate data, especially if the failure modes are assumed to be at the component or part level. Generating the probability of occurrence assumptions at the system or subsystem level presents a little bit of a challenge since failure rate databases do not exist. In this case a combination of the failure rate database at the component level with a parts count type approach at the system or subsystem level could be used to generate the assumptions. As long as this approach can be tied back to actual data of previous systems, subsystems, components, or parts, the probability of occurrence assumption can be made in a traceable manner.

The final assumption of the Hall model is the fix effectiveness factor. This assumption represents the percent reduction in the probability of occurrence of a failure mode if a redesign were to occur for that failure mode. Thus the fix effectiveness factor comes into play after a failure mode has been observed and preventative action is taken prior to the next launch. Similar to the γ_k parameter in the Morse model, the fix effectiveness will be 1 for the case where the failure mode is completely eliminated from the system and 0 when no fix is implemented. This assumption is the least traceable of the parameters contained within the Hall model. As discussed above, it is difficult to pinpoint a value for this parameter even with a plethora of historical data. In the absence of any data at all this assumption must be produced via a best guess or ask the expert type approach, which is less desirable in terms of traceability.

The first "results" from Experiment 1 can now be derived from the qualitative assessment of the traceability of the model parameters. In analyzing the assumptions for both the Hall and Morse models it was shown that the models share three common parameters. These three parameters are the number of failure modes, the probability of occurrence of these modes, and the fix effectiveness factors. As discussed previously, these assumptions are thought to be very traceable compared to the rest. In Section 3.3.2.2 a brief example problem was given, which illustrates how the number of failure modes and probability of occurrence values can be derived from detailed system analysis or previous system data. This is especially true when considering parts or components, which benefit from the existence of failure rate databases.

The remaining parameters from the growth models are only included in the Morse model. Parameters such as the probability of anomaly detection and the probability of anomaly reporting are not measurable in practice. Therefore they can be considered as non-traceable. Due to the fact that the Hall model excludes these parameters it was concluded that the Morse model will be less traceable. The result of the qualitative assessment of the model parameters therefore states that the Hall model performs better in terms of assumption traceability.

3.4.1.2 Model Accuracy Testing

The traceability of the model assumptions will ultimately have an effect on the accuracy of the resulting reliability estimates. Therefore, in addition to the assessment of traceability a quantitative measurement of the model accuracy is required for Experiment 1. This accuracy measurement will allow for the comparison of both of the model outputs versus previous vehicle data. As a result the setup of the model parameters can be further assessed and the model that most well represents the actual data will be chosen.

To begin the model accuracy assessment of Experiment 1 previous data for comparison is required. Due to the fact that the growth models will be applied to a launch vehicle design problem, previous launch vehicle data was desired. Ultimately two launch vehicles were identified for use in Experiment 1. These vehicles were chosen because of their lengthy launch history and the availability of reliability growth data. The first vehicle is the Russian Soyuz launch vehicle.

For the purpose of the comparison of reliability growth models, the Soyuz vehicle is a perfect candidate. This is due to its extensive launch history through which the vehicle architecture remained unchanged. The Soyuz vehicle's heritage can be traced back to the R-7 rocket developed by the Soviet Union in the 1950's [90]. As can be seen in Figure 18 the architecture of the Soyuz is nearly identical to the R7 with the exception of the addition of an upper stage. This is advantageous for applying the reliability growth models in Experiment 1. The models can be setup to represent the Soyuz vehicle architecture and the output can be compared to the entire reliability growth history of the Soyuz family of vehicles.

The Soyuz launch vehicle itself consists of three stages. The first stage consists of four symmetrical liquid rocket boosters (LRBs) surrounding the central "second stage" [90]. Each of the boosters and the second stage have one liquid rocket engine with four separate combustion chambers, which utilize liquid oxygen and kerosene as propellant [90]. The third stage sits inline atop the second stage and contains another single liquid rocket engine with four combustion chambers [90]. The third stage engine also uses a kerosene propellant with liquid oxygen. As seen below, the right side of Figure 18 shows a picture of the Soyuz launch vehicle in unmanned configuration.

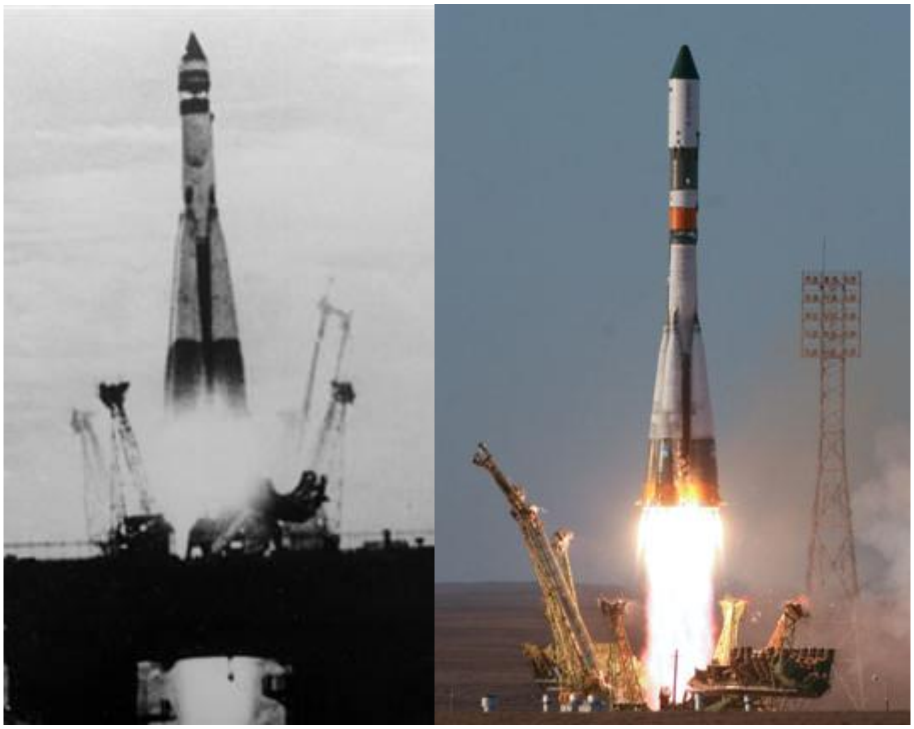


Figure 18: Soyuz launch vehicle (right) compared to the R-7 vehicle (left) [122, 125]

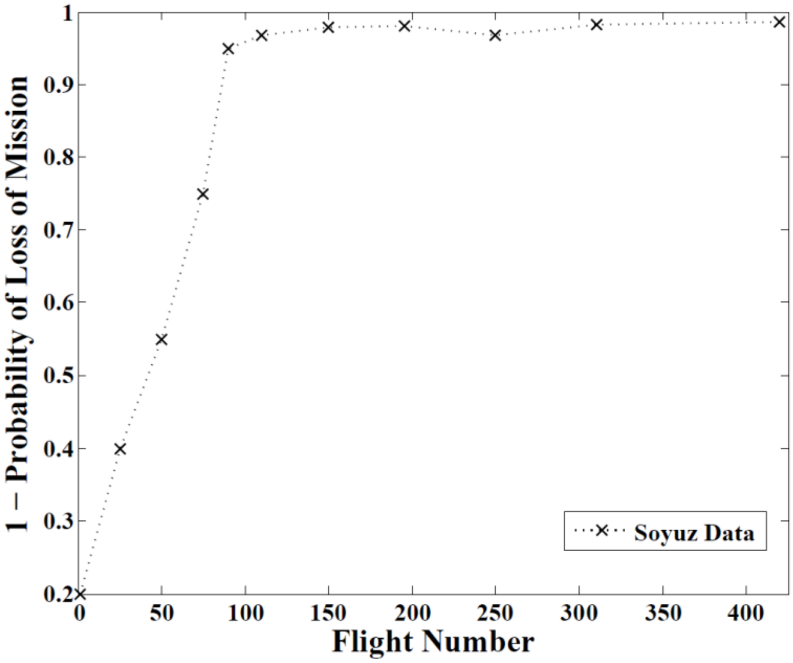


Figure 19: Soyuz launch vehicle reliability growth

The reliability growth data for the Soyuz vehicle that will be used in Experiment 1 can be found in reference [62]. This data is also cited and re-illustrated in references [58] and [108]. Figure 19 shows a re-creation of the Soyuz reliability growth data from the references above. As can be seen in the figure the data shows reliability growth over approximately 425 flights.

The second vehicle that will be used for model accuracy checking is the Space Transportation System (STS), also known as the Space Shuttle. The Space Shuttle was chosen because it benefits from a relatively long flight history (> 100) with the same vehicle architecture. In addition, a vast amount of documentation in regard to the design, test, and operation of the Shuttle is available.

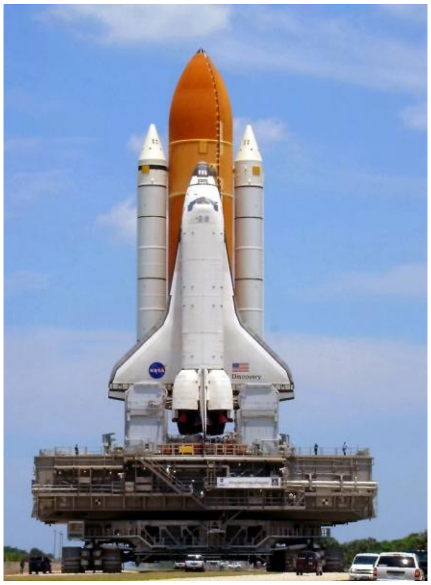


Figure 20: Space Shuttle Discovery on mobile launch platform [123]

The Space Shuttle consists of four primary elements including the orbiter, external tank, and two re-usable solid rocket boosters. On launch the solid rocket boosters (SRB) are ignited along with three Space Shuttle Main Engines (SSME) housed in the orbiter. These liquid engines burn a combination of liquid oxygen and liquid hydrogen, which is stored within the external tank. Figure 20 shows the fully assembled STS vehicle on a mobile launch platform. The black nozzles of the SSMEs can be seen in the foreground just aft of the orbiter vertical stabilizer. The external tank stands out prominently in orange with an SRB attached to either side.

The STS system benefits from the availability of a vast amount of technical data, which includes reliability growth data. Beginning in the 1980's and carrying through the retirement of the vehicle a highly detailed probabilistic risk assessment (PRA) of the Space Shuttle was performed [129]. This assessment can be considered one of the most detailed and accurate ever carried out for a launch vehicle program. Near the end of the Shuttle program the PRA was used to demonstrate the reliability growth that was achieved throughout its operating life. References [18], [78], and [79] provide data tables illustrating the reliability growth of the STS in terms of decreases in the probability of loss of crew. These tables provide the mean LOC probability along with confidence bounds in the form of 5th and 95th percentiles. Figure 21 shows the graphical form of the data including the percentiles. The data seen in the figure illustrates the decrease in the probability of LOC between STS-1 and STS-133.

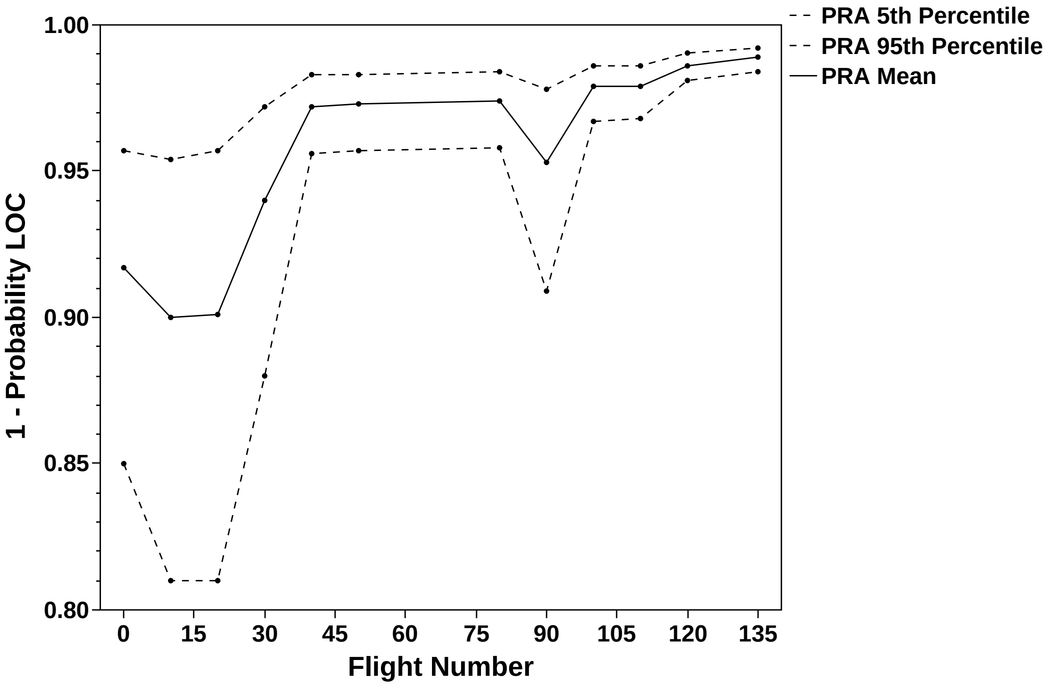


Figure 21: Reliability growth curve from detailed STS PRA

After identifying two vehicles for use in Experiment 1 the assumptions for both the reliability growth models must be setup appropriately. In order to fully test the accuracy of the growth models it will be assumed that the two identified vehicles are new designs, which are not comparable to any historical vehicles. This is an appropriate assumption considering the heritage of the Soyuz tracing back to one of the first launch vehicles ever operated. Also, the Space Shuttle at the time of its conception was a completely unique concept that had not been attempted before. In order to avoid biasing the results, previous operational data, other than the reliability growth data presented above, will be avoided for both these vehicles.

First, the assumptions for the number of failure modes will be completed. Since comparisons or design data from previous systems do not exist, a simple parts count type approach is appropriate for generating this assumption. The data retrieved for the STS shows probability of LOC, therefore the failure modes to be considered for this vehicle will stem from "parts" contributing directly to an LOC event. Since the Soyuz data represents probability of LOM, only the "parts" that will lead directly to LOM will be counted. Utilizing this very basic approach for generating the number of modes assumptions represents a worst case scenario in terms of model prediction accuracy.

The number of failure modes for the Soyuz vehicle will be considered first. As shown in Figure 18 the Soyuz vehicle has four liquid boosters surrounding a liquid core stage that ignite upon launch. Each booster contains liquid oxygen and kerosene to be burned by its single engine. The core stage also burns liquid oxygen and kerosene using a single liquid engine with four combustion chambers. The final Soyuz stage to consider is the upper stage. There are multiple variants of the upper stage that have been used throughout the program history [99]. The most common architecture is a single engine liquid oxygen and kerosene upper stage [99].

In order to identify the appropriate number of "failure modes" for the system level assumptions it is important to refer back to the discussion in Section 3.9. This section discusses the development of the reliability growth model assumptions and addresses the approach for generating the number of failure modes assumption without historical data. It is expected that a parts count type approach is appropriate in this case as long as the "parts" that are counted contribute directly to the top level event of interest.

For the Soyuz case the top level event of interest is a loss of mission. With this in mind, the count of "failure modes" for the system level should be equivalent to the number of events that contribute directly to a loss of mission. This can be illustrated using a notional fault tree, which identifies all of the primary subsystem failures that would contribute directly to an LOM. Figure 22 gives such a tree for the Soyuz vehicle. Note that the liquid rocket boosters are labeled as LRB with a corresponding number.

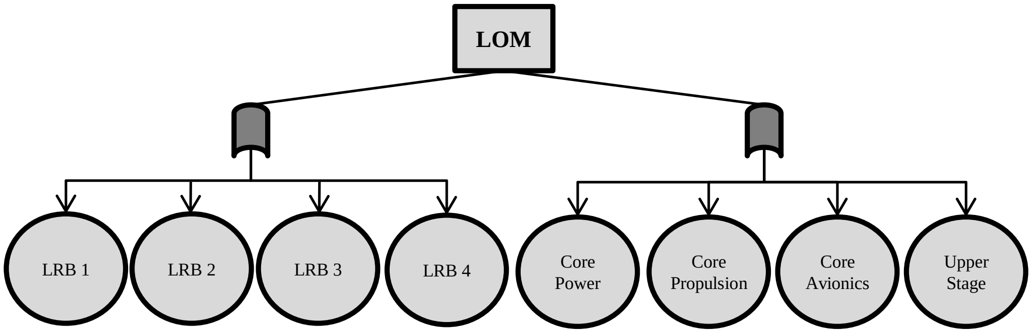


Figure 22: Notional Soyuz high level fault tree

From the figure above, eight possible high level failure modes can be identified for the Soyuz vehicle. The core stage has been broken out into propulsion, power, and avionics because the power and avionics subsystems are considered to be vital to a successful ascent. For the system level reliability growth assumptions, eight modes are listed in the figure above. In addition to these eight it is possible to also include a common cause type failure for the liquid boosters as well as an upper stage avionics or power system failure. The latter two failures are especially important to note because the upper stage operates under its own power and guidance to reach orbit.

Considering these extra failure modes, a range between 8 and 14 will be used for the Soyuz system level assumptions.

It is important to note that the level of application of the growth model is a determining factor in how the number of failure modes assumption is produced. Due to the fact that the liquid rocket boosters are considered as subsystems, they are not broken down any further to identify more specific failure modes. If the growth model were to be applied at the subsystem level, each of the basic events seen in Figure 22 would be broken down into one more level of detail. This process will be discussed in Section 3.5 and performed in the example problem presented in Section 4.2.

Next, the number of failure modes assumption for the STS vehicle needs to be generated. The STS consists of four primary elements, the orbiter, solid rocket boosters, and external tank. The orbiter houses three primary subsystems that are essential to the success of an STS launch, the primary avionics, power, and Space Shuttle Main Engines.

For the STS assumptions the high level fault tree given in Figure 31 from Section 4.2 can be used. In this figure, the SRBs and external tank are left as basic events directly below the top level event. The orbiter has been broken down further because of its role in the success of a given launch. As with the Soyuz core stage above, the avionics housed in the orbiter are critical to a successful ascent of the vehicle. In addition, the SSMEs provide the sole source of thrust for the vehicle after the SRBs are jettisoned. For these reasons the breakdown of the orbiter was considered reasonable for generating the system level reliability growth assumptions. The growth model for the STS vehicle will therefore use a range of 6 to 12 failure modes for the system level.

The next assumption to be considered is the probability of occurrence of the identified failure modes. This assumption is much more difficult to derive without historical data for comparison. However, both of the authors of the growth models being considered give some insight into estimating this parameter. First, in development of the model, Morse identifies three different trigger frequency groups for the system failure modes [108]. Within these frequency groups Morse identifies broad ranges for both probability of trigger and conditional probability of loss of mission. It is important to note that the Morse model includes two parameters that determine the probability that a failure mode will cause LOM, while the Hall model only uses one. The probability of occurrence parameter in the Hall model is equivalent to the product of the probability of trigger and conditional probability of LOM within the Morse model. Table 4 displays the product these parameters from the Morse model, using the ranges given by Morse in [108]. From this table the overall range in probability of occurrence of LOM can be taken as 0.05% to 71% as suggested by Morse.

Table 4: Probability of LOM for three failure frequency groups

Frequency Group	Probability of LOM
High	18% - 71%
Medium	0.25% - 7.5%
Low	0.05% - 1.5%

Note that in addition to identifying the ranges for probability of LOM for each group, Morse identifies a range for the number of expected modes that fall within each frequency group. These ranges are 0 - 5 modes for the high frequency group, 1 - 12 modes for the medium frequency group, and 2 - 20 for the low frequency group. Considering the number of modes for the Soyuz and STS vehicles, it is expected that most of the modes will fall within the low frequency group with one or two landing in the medium or high groups.

Next, the probability of occurrence assumptions presented by Hall will be considered. In references [75], [76], and [77] Hall represents the probabilities of occurrence of the system failure modes using a Beta distribution. He also gives procedures for estimation of the Beta shape parameters based upon given failure data. These procedures are set up to estimate the Beta shape parameters for two cases. The first case is where the number of failure modes, k, is known. The second case allows the number of failure modes to approach ∞.

Hall demonstrates these estimation procedures using assumed one-shot systems such as an air-to-ground missile [77]. From the references above, multiple different shape parameter combinations for complex one-shot systems such as missiles can be identified. Table 5 lists some of the derived Beta parameters from [76] and [77] along with the distribution mean and maximum values. These distributions can be compared to the ranges given by Morse in Table 4 above.

Table 5: Beta parameters for probability of failure mode occurrence

Beta Parameters (α, β)	**Mean**	**Maximum**
0.19, 23.31	0.008	0.3
0.36, 14.99	0.023	0.4
0.19, 8.03	0.022	0.52
0.22, 8.75	0.024	0.54

Table 5 displays multiple sets of Beta parameters given by Hall for complex one-shot systems. These distributions show a mean probability of occurrence of 0.8% to 2.3% and maximum values of 24% to 54%. In comparing these values to Table 4, all of the maximum values fall somewhere within the high frequency group and the mean values fall mostly within the low frequency group. Also, the number of failure modes within each category provided by Morse would suggest a similar distribution shape to what is given by Hall. As discussed above, Morse suggests that the number of modes within the three frequency groups is skewed towards the low frequency group. Therefore, a majority of the failure modes would fall between 0.05% and 1.5%

probability of occurrence. If a random draw was taken using the Hall distributions in Table 5 a majority of the cases would also fall within a similar range.

Due to this similarity in shape, a Beta distribution was determined to be appropriate for the probability of occurrence assumptions. Since Hall provides Beta parameters already, a distribution from Table 5 was selected for application in Experiment 1. The parameters that will be used are the last in the list in Table 5, which were selected because they produced the largest maximum value for probability of occurrence. This was desired because the Morse assumptions indicate a maximum of around 70% for probability of occurrence. This distribution also provides a mean of around 2.5% with approximately 50% of the points within the low frequency group identified by Morse.

Next, the assumptions for fix effectiveness factors can be considered. As discussed in Section 3.3.1.5 the fix effectiveness factors represent the percent reduction in the probability of occurrence of a failure mode, given that the mode has occurred and a fix has been implemented. This parameter is equivalent to the γ_k parameter within the Morse model, which represents the probability that a defect is eliminated from the system given that it has been properly detected and reported. For this experiment the same values will be applied for both the fix effectiveness factors and the γ_k parameter in the Hall and Morse models, respectively. This will avoid potential bias within the model results.

In Section 3.9 the generation of the fix effectiveness assumption was discussed in more detail. It was identified that in practice, the fix effectiveness factors are difficult to derive from vehicle data. Section 3.9 concluded that the most appropriate manner of producing these assumptions was via expert judgment. This conclusion is supported by Hall during the development of his growth model. Hall states that the fix effectiveness factors are typically assessed via expert judgment and are assigned during failure prevention review boards [77]. In lieu of a board of experts, values from the literature from Hall and Morse can be utilized for this experiment.

In reference [108] Morse states that the fix effectiveness or γ_k parameter is typically very high for launch vehicles. This is due to the large amount of post flight data analysis that is typically performed after a launch. The large expense of a launch vehicle failure demands that any and all defects that have been detected within the system should be investigated and potentially mitigated. Morse calls out a range of between 75% and 90% for the γ_k parameter [108].

As mentioned previously, Hall acknowledges that the fix effectiveness factors are typically determined via expert judgment [75, 77]. In reference [77], he gives a table of fix effectiveness factors from an unspecified air-to-ground missile program. These factors are said to have been developed during a failure prevention and review board for the program [77]. The table presented by Hall shows a range of 70% to 95% for the fix effectiveness factors of the missile system.

From the ranges given by Hall and Morse a range for application in Experiment 1 can be derived. As discussed previously, the fix effectiveness will be greatly affected by the rigor of post flight data analysis and designer experience. The fix effectiveness may also be positively impacted by a long flight history, which allows the designer to assess the vehicle many times within its operating environment. The more detail the designer has about the operation of the vehicle systems the better equipped they will be to completely correct defects that surface in the future. Due to the fact that the Soyuz and STS vehicles are both manned a large amount of post flight data analysis can be expected in each program. In addition these vehicles benefit from a relatively long flight history using the same vehicle architecture. Therefore, we would expect to achieve a very high value for fix effectiveness. The range of fix effectiveness used for Experiment 1 will then be set based upon the top end of the ranges stated by Hall and Morse. Experiment 1 will utilize a uniform distribution for the fix effectiveness factors ranging between 90% and 99%.

After determining the fix effectiveness assumption, all of the parameters for the Hall model have been covered. The remaining assumptions that need to be addressed are all from the Morse model. The first of these remaining assumptions is the minimum probability of failure for the vehicle. This parameter represents the maximum reliability of the vehicle, which will have a great effect on the accuracy of the growth estimate. As discussed in Section 3.9 this assumption is difficult to determine for a specific vehicle. The only way to generate this assumption is to determine the maximum reliability of previous vehicles, which may or may not be similar to the vehicle being analyzed.

Morse states that the historical best for this parameter is a probability of failure of 1 in 200 at 200 flights [108]. For the STS vehicle, this value will be used directly because the launch history only includes 135 flights. This suggests that the vehicle could approach a probability of failure of 1 in 200 after 70 additional flights. The Soyuz vehicle has a much longer history of around 425 flights, which will require a slight adjustment in the minimum probability of failure. Since the Soyuz vehicle has over twice the number of flights as what was quoted by Morse as the historical best, a minimum probability of failure of 1 in 400 will be assumed.

The remaining two parameters of the Morse model are; the probability that an anomaly is observable given that it has occurred and the probability that the anomaly is noticed and reported. In Section 3.9 it was pointed out that these assumptions are particularly difficult to determine, even if a plethora of historical data is available. The probability that an anomaly is reported is greatly affected by both the rigor of the post flight analysis and the analyst carrying out the review. For these reasons the ranges suggested by Morse will be used in Experiment 1. The range given for the probability of anomaly detection is 75% to 95% while the range for the probability of anomaly reporting is 75% to 90%.

Now that all of the assumptions for both the growth models have been determined, the models can be implemented for each vehicle. The equations given by Morse [108] and Hall [75] were thus coded in Matlab in order to carry out the analysis. A Monte Carlo simulation using both models was carried out for the Soyuz and STS vehicles utilizing the assumptions laid out above. In order to compare the output accuracies, the model results were compared to the retrieved data for each launch vehicle. A total sum of the error was taken across the flight histories in order to make a quantitative comparison between the models. This process will be discussed in more detail in the next Section, Section 3.4.2. Table 6 below gives a summary of the assumptions that have been determined above.

Table 6: Reliability growth assumptions for Experiment 1

Parameter	Soyuz Value	STS Value
Number of Modes	8 - 14	6 - 12
Probability of Occurrence	Beta(0.22,8.75)	Beta(0.22,8.75)
Fix Effectiveness	90 - 99 %	90 - 99 %
Observable Anomaly	75 - 95 %	75 - 95 %
Reported Anomaly	75 - 90 %	75 - 90 %
Number of flights	425	135

3.4.2 Experiment 1 Results

Section 3.4.1.2 presented two vehicles for use in testing the Hall and Morse model accuracies. After identifying the STS and Soyuz vehicles the growth model assumptions were developed near the end of the section. Using these assumptions both growth models were run within Matlab. For each model, reliability distributions at each step in time were output, which allowed for the generation of the model mean and percentiles to compare to the historical data. In order to generate these distributions at each step in time a Monte Carlo (MC) simulation was run for each step using a random draw from probability distributions created from the assumptions in Table 6.

A quantitative error metric was then introduced in order to compare the Hall and Morse models. This metric was calculated as the total sum of the error of the models at each step in time. The error was defined as the difference between the model mean and the mean from the historical data. These errors can be seen for both the Soyuz and STS vehicles in Table 7. This table quickly shows that the Morse model performs worse in terms of total error. The Hall model has nearly one half of the total error of the Morse model for both vehicles. Based upon this observation it is expected that the Hall model results will more closely approximate the actual vehicle data.

Table 7: Reliability growth assumptions for Experiment 1

Model	**STS Total Error**	**Soyuz Total Error**
Hall Model	0.3897	0.9601
Morse Model	0.5344	2.037

After observing the generic error data for both models and both vehicles, the full output of the models can be examined. Figure 23 and Figure 24 display the results from the Hall and Morse models respectively, versus the given Soyuz reliability growth data.

The first figure, plotting the Hall model results, shows a decent agreement between the predicted and actual reliability values. The Hall model seems to capture the correct curvature of the growth history, but it does over estimate the reliability of the vehicle during the early flights. This over estimation occurs primarily between flight 0 and flight 75, where the Hall model results begin to fully encompass the actual data. Although this over estimation is large at certain points, the model prediction starts to converge towards the actual value during the later portions of the flight history.

The over prediction of the early reliability of the Soyuz by the Hall model becomes less worrisome after analyzing the Morse model results. As can be seen in Figure 24, the Morse model grossly over predicts the Soyuz reliability for nearly the entire flight history. In this case the over prediction spans from flight 0 all the way to flight 425. These results show that the Morse model was not able to accurately predict the growth trend of the Soyuz vehicle. Therefore, the Soyuz results for Experiment 1 are not a promising start for the Morse model.

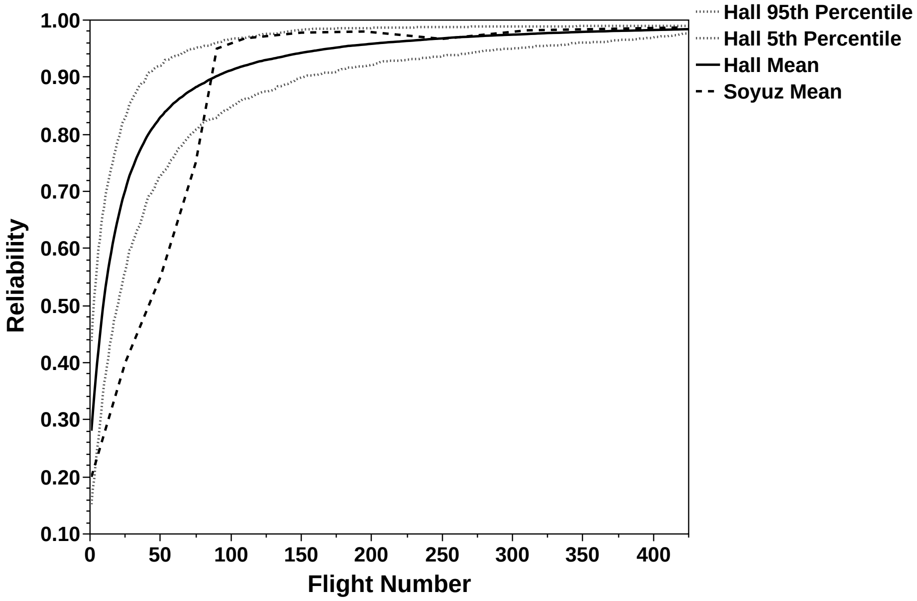


Figure 23: Hall model growth prediction versus Soyuz data

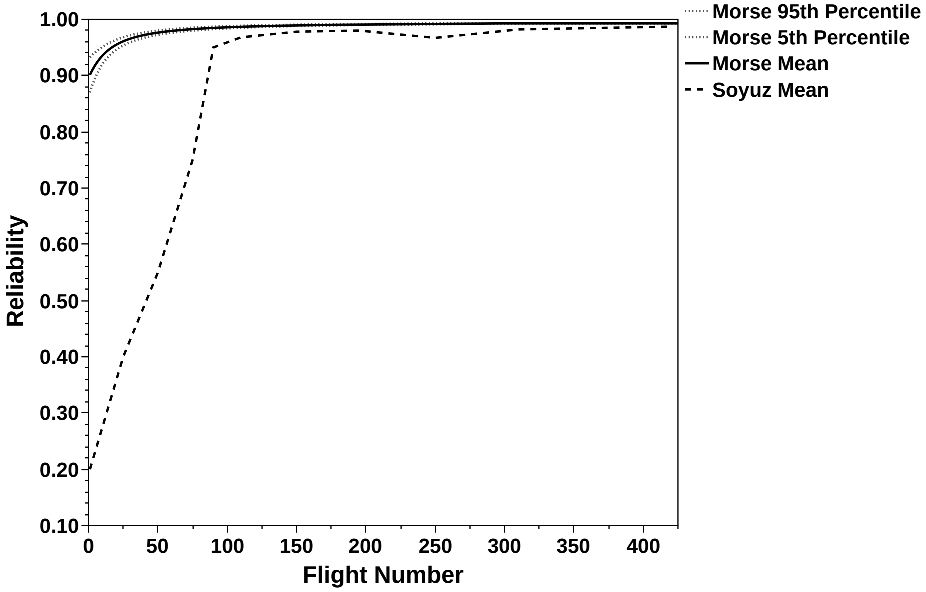


Figure 24: Morse reliability growth model prediction versus Soyuz data

After considering the Soyuz reliability growth predictions, the Hall model can be considered the leading candidate for application in the CONTRAST method. The STS vehicle results will ultimately determine if the Hall model will be selected. Figure 25 and Figure 26 plot the predicted versus actual reliabilities for the STS vehicle.

In Figure 25, the Hall model reliability prediction is plotted versus the PRA data for the STS. Note that the STS reliability data contains not only a mean value, but a 5th and 95th percentile value as well. Similar to the Soyuz vehicle, the Hall model captures the reliability growth trend of the STS very well. However, the STS vehicle prediction actually performs better because there is no longer any over prediction during the early flights. As can be seen in the figure, the Hall model mean value tracks very well with the actual PRA data and ends up slightly higher than the actual STS data at the last flight. The 5th and 95th percentiles from the Hall model have a fairly wide range, but are able to encompass nearly all of the PRA data.

Figure 26 shows the Morse reliability growth prediction versus the same STS PRA data. This figure shows a similar trend to the Soyuz prediction from the Morse model. In this case, the Morse model has again over predicted the reliability of the vehicle for the entire flight history. Although the predicted data seems to capture the general shape of the actual reliability growth trend, the values for reliability are well above the actual.

After considering the Soyuz and STS reliability growth results from the Hall and Morse models, the appropriate growth model has become very apparent. For both of the vehicles the Hall model tracked very well versus the actual data. The Morse model on the other hand was not close to the actual data as it was drastically over predicting the reliability of the vehicles over their entire flight history. From these results it is clear that the Hall model will be able to provide more accurate reliability growth predictions. In terms of relative accuracy, the Hall model can be deemed the most appropriate for application in the CONTRAST method.

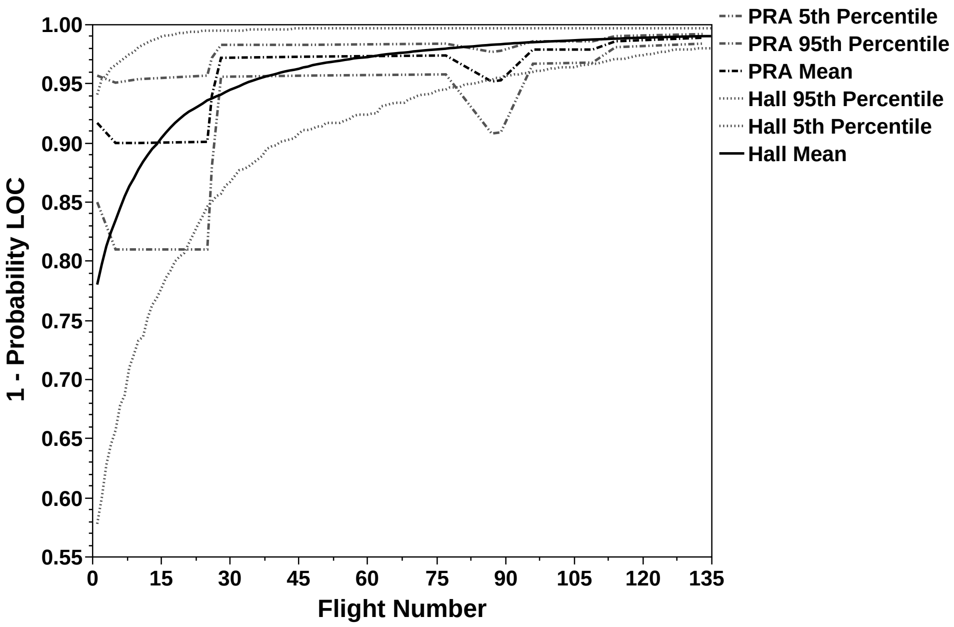


Figure 25: Hall reliability growth model prediction versus STS data

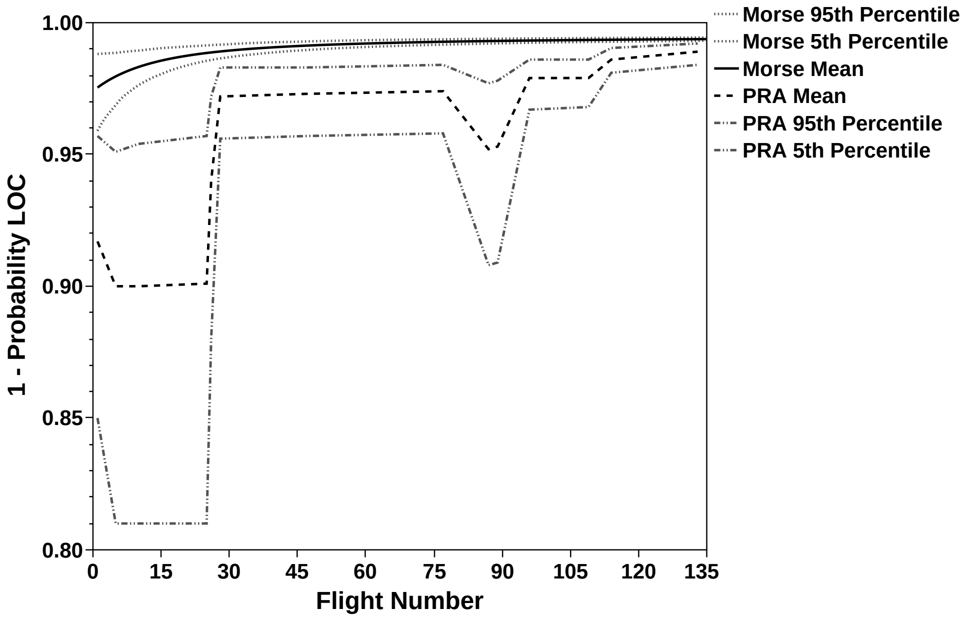


Figure 26: Morse reliability growth model prediction versus STS data

3.4.2.1 Conclusions

Experiment 1 tested two reliability growth models, which were potential candidates for application in the CONTRAST method. Both the traceability of the assumptions and the relative accuracy of the model outputs were tested. As discussed in Section 3.4.1.1, the Hall model was deemed to be more desirable in terms of traceability. This is primarily due to the large number of parameters that must be estimated in order to run the Morse model. In Section 3.4.2, the relative accuracy of the growth model predictions was tested versus two actual launch vehicles. From the results in this section it is clear that the Hall model will perform better in terms of prediction accuracy.

Due to the fact that the Hall model performed better for both of the major selection criteria, it can be officially chosen for use in the CONTRAST method. This conclusion is in line with the original hypothesis, which identified the Hall model as the most desirable. Therefore, hypothesis 2 can be accepted based upon the results of Experiment 1.

3.5 Research Question 3: Level of Application

In the previous section a reliability growth model was selected for application in the CONTRAST method. Growth methods were chosen because they generate reliability estimates as a function of time. This type of output was identified as the most desirable in Section 3.1. Although the method selected in the previous section produces the desired output, the application of the growth model must be determined. These models can be applied at many different levels of characterization of the system, which will have large implications in terms of the availability and traceability of the information used to generate the model assumptions. To determine the appropriate level of application research question 3 was posed.

Research Question 3

What level of characterization is appropriate for the application of a reliability growth model during conceptual design?

The levels of characterization of a system are related to the hierarchical decomposition of the system from the top level down to its parts [15]. An example decomposition from reference [15] will be used for this discussion. The eight level characterization seen in Table 8 shows the breakdown of a system from the top level down to the specific material of each part. At first glance there are seven possible levels at which the growth model could be applied, which excludes the lowest material level. However, not all of these levels are applicable early in the design process.

Table 8: Eight-level system characterization

Level	**Characterization**
0	System
1	Subsystem
2	Major Assembly
3	Assembly
4	Subassembly
5	Component
6	Part
7	Material

In order to address research question 3 the levels of characterization can first be compared to the generic design process from Section 2.1. The typical launch vehicle design process can actually be equated to a descent through the levels of characterization seen in Table 8. During the early pre-conceptual and conceptual design phases focus is placed on identifying a baseline vehicle architecture that can meet the specified program requirements. At the end of the conceptual phase a baseline vehicle architecture has been selected, which includes a description of the vehicle architecture along with generic descriptions of its subsystems. The early design phases therefore only address the top-most levels of characterization.

The preliminary design phase introduces increased fidelity analysis for all of the subsystems within the vehicle. The goal of this phase is to solidify the set of system and subsystem design specifications, which may result in the production of engineering test articles. In producing such specifications more details will be determined for the assembly and component levels of characterization.

The final phase, detailed design, defines the detailed specifications of all hardware within the system. During this phase built-to specifications are produced, which are used to fabricate more detailed test articles or actual flight hardware. The detailed design phase therefore fills in the remaining information regarding the lowest levels of characterization including part and material.

The comparison of the design phase to the levels of characterization shows that the level of application of the model will have a large effect on the amount of information required to define the model parameters. At the lowest levels not only individual part or component specifications are needed but any interactions and relationships between them as well. However, there may be an added benefit to using lower levels when considering the availability of data for producing the assumptions.

The natural starting point for producing the reliability growth model assumptions is to look at data from a previous similar system. For launch vehicles this data includes any test or flight history from a system with a similar architecture. This criterion of similarity will actually be easier to satisfy at the lower levels of characterization. To illustrate this point, consider the production of system and subsystem level growth assumptions based upon historical data.

In order to set the system level model assumptions based upon historical data, the new vehicle must first be compared to previous vehicles to find a suitable surrogate. As discussed in Section 3.2.1.7, MIL-STD-756B gives guidelines for the validity of these types of comparisons. Therefore, the success of this approach is reliant on the existence of previous vehicles that are similar to the new vehicle. In addition, the previous vehicles must have a sufficient operating history in order to prove a certain level of reliability achievement. This point is especially relevant when considering launch vehicles. Although the U.S. has had many different launch vehicle programs in the last 60 years, not all of them can be utilized for comparisons. This is due to either insufficient amounts of data, or the complete lack of data altogether.

Programs with insufficient data are ones in which the vehicle made it through the design process only to get canceled after a small number of test flights. Two examples of this type of program are the Blue Scout and the Ares I. Blue Scout flew two test missions and a failed satellite launch attempt in 1961, only to get canceled before the production of more operational vehicles [93]. Ares I is a more recent example, which flew its first test flight in 2009 before being canceled in 2010 [1]. In both these examples, the vehicles did not fly enough to demonstrate an achieved level of reliability.

There are many more programs that lack reliability data altogether. These programs are ones in which the vehicle did not perform any test flights before cancellation. Examples of this type of program are the Navy Earth Satellite Vehicle Program in the '40s, Project Orbiter in the '50s, Orbital Express in the '90s, and the Ares V in 2010 [93]. Each of these programs went through the early design phases, but were canceled prior to flight testing.

As illustrated by these examples, generating the growth model assumptions at the system level may be difficult due to the limited amount of historical data. This limitation stems from the fact that the entire system must demonstrate a certain level of reliability achievement for the data to be viable for comparison. It is expected that more data will be available at the subsystem level due to the amount of testing that is performed during a typical vehicle program. Although the full vehicle may not have flown, the engines, structures, separations systems, etc. all may have undergone extensive testing. This testing is sufficient for producing reliability data that can be used for comparisons.

For example, before the first flight of STS, the Space Shuttle Main Engine was estimated to have undergone approximately 70,000 seconds of testing at rated power level [11]. If the program had been canceled at that point, sufficient data for comparison to the SSME would have been available. However, no data would have been available for the STS as a whole, which would remove it from consideration for any system level comparisons.

In addition to testing that occurs during development programs, subsystem data benefits from technology demonstration programs as well. Many of the NASA X programs did not produce operational vehicles, however, they did serve as successful test beds for the demonstration of new subsystem technology [113]. Other technology development programs, such as the Air Force integrated high-payoff rocket propulsion technology program or the NASA Space Launch Initiative, produced many different engine technology demonstrations [141]. Although a majority of the engine development programs were canceled, many of them underwent physical testing, from which data could be derived for reliability comparisons.

Through the previous discussion two observations can be drawn regarding the level of application for the growth models. First, the highest levels of characterization are more desirable in terms of the information required to generate the model assumptions. At the lowest levels the exact type and number of components or parts must be known along with their associated specifications in order to generate the model assumptions. Second, considering the availability of previous reliability data shows an opposite trend. If the growth model assumptions are generated based upon comparison to historical vehicles it is expected that more viable data will be available at the lowest levels. This is primarily due to the amount of sub-scale testing that typically occurs during a launch vehicle program.

With these observations in mind it can be concluded that the appropriate level for application of the growth models will find a balance between the required detail and the availability of information for comparison. If no data exists for comparison the assumptions for a new system must rely solely upon judgment, which will introduce error into the output. On the other hand, if the level of application requires too much detail regarding the specific components of the system the assumptions cannot be produce using the information available during conceptual design.

After noting the trade-off that exists between data availability and required level of detail, some typical design trades can be considered. This will help identify the level of detail required to capture any relevant trades that could be considered during conceptual design. It will also clarify the ability of the model to capture certain architecture effects when applied at different levels of characterization.

Typical design trades and architecture upgrade options can be identified from existing vehicles as well as current design work. Vehicles such as the STS utilized the same architecture throughout the entire program, however, incremental upgrades were performed on some of the subsystems. For example the oxidizer and fuel turbopumps within the SSME were upgraded at separate points during the program [78]. Other upgrades to the vehicle included changes in the thermal protection system of the solid boosters and a new application process for the external tank foam [78]. Considering these types of incremental upgrades would require the ability to support trades at the assembly level, which is directly below the subsystem level.

Another list of possible architecture trades can be derived from the concept of launch vehicle families. For example, the Atlas V and Delta IV vehicles each have multiple different variants, which can be selected by the customer based upon the payload and desired orbit [160, 161]. The Delta IV vehicle utilizes a common core element, which makes up the first stage of every variant [161]. In addition to the first stage, strap-on solid boosters can be included on the vehicle with the Delta IV heavy variant utilizing two common core elements as boosters [161]. The Atlas V family contains a number of variants that differ based upon the number of strap-on solid boosters and the number of engines in the upper stage [160].

The architecture changes within the Atlas and Delta vehicle families represent trades at a higher level than the STS example from above. Trades such as number of strap-on boosters or number of upper stage engines represent subsystem level decisions. These trades are more typical of the studies performed during conceptual design. For example consider the Space Launch System (SLS), which is currently being developed by NASA. During the early design phases of the SLS, studies were performed to identify the appropriate liquid engine and number of engines for the core stage of the vehicle [86]. In addition, potential upgrades for the upper stage and boosters are being considered [32, 33, 151]. These options represent trades at the subsystem level.

During conceptual design the subsystem level trades illustrated by the Atlas and Delta vehicle families and the SLS design work will be of most interest to the reliability analyst. Trades at the assembly level may be of some interest if heritage hardware that may require upgrades for future use is being considered. However, considering trades at levels below the assembly level may not be practical and will not necessarily offer any added value to the design process.

After identifying subsystem and assembly level trades that would be of interest during conceptual design the levels of application of the growth model can be considered. This thought experiment will clarify the approach needed to capture such trades when applying the model at different levels of characterization. The application of the model at the system level will be considered first.

In Experiment 1 the growth models were applied at the system level and a basic parts count approach was used to generate the number of failure modes assumption. This approach was applied at the next lowest level of characterization, which counted the subsystems within the vehicle. The modes included in the growth model therefore represent the subsystems and their associated probabilities of failure.

In order to perform subsystem level trades using the system level growth model the modes assumptions can be adjusted accordingly. For example, addressing a trade between number of engines or boosters would only require a change in the number of modes included in the model. If multiple different alternatives were being considered for a specific subsystem, the probability of failure assumption would need to be used to differentiate between the options. However, certain configurations of the subsystems cannot be captured using the system level model. The formulation of the growth model inherently assumes that each of the failure modes are independent and in series. Therefore, redundant subsystems cannot be captured directly using the system level assumptions. In addition, options such as engine out capability will not be captured.

Applying the growth model at the second level of characterization, subsystem, will help alleviate the limitations of the system level model. In this case the parts count approach identifies the key assemblies within each subsystem. For example, the parts count for a liquid engine would include items such as the fuel and oxidizer turbopumps and the primary combustion chamber. Subsystem level trades such as number of engines or boosters can now be captured by adjusting the number of implementations of the model. Capturing differences between subsystem options can also be done in more detail as both the number of modes and probability of occurrence distributions can be adjusted.

The application of the model at the subsystem level also introduces the ability to consider assembly level trades. These trades will include incremental updates to the vehicle subsystems similar to the STS upgrades discussed previously. For example, to represent a potential upgrade to an engine turbopump the probability of occurrence value for the specific mode can be adjusted. Additionally, the number of modes can be adjusted to explore trades between single and multiple turbopump configurations. The subsystem level model can explore assembly level trades, but it is important to note that the assemblies within the model are all assumed to be in series. Therefore, redundancy at the assembly level cannot be captured using this level of application.

Applying the growth model at the next lowest level would allow for the capture of redundancy at the assembly level. However, this level of application begins to require more detail in order to assess trades at the system and subsystem levels, which are of most interest during conceptual design. In applying the model at the assembly level, each subsystem will represent some number of model instances. A subsystem with 8 primary assemblies for example, will require a set of assumptions for 8 growth models. Since the models represent each assembly, the failure modes within the models pertain to individual components. This clearly illustrates a potential issue with applying the growth model at the assembly level or lower. A vast amount of information would be required to setup the assumptions for every component within the system.

Assuming all of the information is available and the growth models are implemented for each assembly, the system and subsystem level trades can be captured. In order to implement these trades however, an additional step is needed. To wrap the assembly level model output all the way to the system level the relationships between the assemblies must be known. Using an additional analysis technique such as a fault tree or reliability block diagram the subsystem and system level reliabilities can be calculated. This means that in addition to the detailed data required to produce the model assumptions for each component, the relationships between the assemblies must be known. Although application of the growth models at lower levels of characterization would capture very detailed design trades, producing the necessary model assumptions is not realistic during conceptual design.

Following the identification of typical architecture trades of interest and a brief assessment of the ability to capture such effects using growth models varying levels of characterization, two primary observations can be stated. First, the highest level of application for the growth models will limit the amount of available historical information for generating the assumptions. Although historical data is not the only reference for assumption generation, it serves as a traceable source that can at least be used to validate the growth model. In addition to this potential lack of data, the system level application is inherently limited in the architecture effects it can capture. Since the growth model assumes that all modes are in series, the effects of subsystem redundancy and engine-out capability cannot be observed.

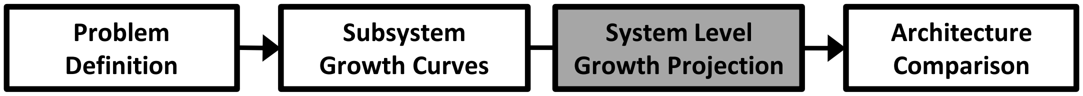
The second observation is in regard to the application of the growth models at the assembly level or lower. Applying the models at these levels will likely improve the availability of historical data for comparison, however, a large amount of information is required regarding the vehicle design. In order to apply the growth model at such a low level, details for every component and the relationships between each assembly are needed. Even if this information was available during conceptual design the application of the growth models at the assembly level will capture more detail than is necessary.

After stating the observations the subsystem level of application appears to be the most appropriate option. The system level of application will not allow for the capture of all relevant architecture trades, while lower levels will provide more information than is necessary. Therefore, the CONTRAST method will apply the growth models at the subsystem level in order to balance the information availability and level of output detail. This conclusion is reflected in the assertion to research question 3.

Assertion to Research Question 3

Applying the reliability growth models at the subsystem level will provide an adequate level of detail to capture relevant architecture trades while avoiding issues with data availability.

3.6 Research Question 4: System Level Estimates



The conclusions drawn from research question 3 resulted in the decision to apply the reliability growth model at the subsystem level. However, the desired output of the method is a reliability growth projection at the system level. The Reliability and Safety Analysis step of the generic decision-making process will therefore be split into two separate steps. The first represents the application of the growth models to the subsystems. The second will require an approach for wrapping the subsystem level growth curves up to the system level. In order to identify an approach for this step research question 4 was posed.

Research Question 4

How can subsystem level reliability growth curves be combined to produce an overall system level growth estimate?

Five potential solutions to research question 4 can be identified. The five possible options to combine the subsystem reliabilities into a system level reliability are: simply multiply the subsystem level estimates across all subsystems, use reliability block diagrams, fault tree analysis, Markov chains, or stochastic Petri nets. In this case, referring back to the derived requirements for successful completion of the main research objective will aid in selecting a technique. From the requirements, stated in Section 2.4, two primary metrics can be derived to facilitate comparisons between the options for research question 4.

First, the method must be capable of analyzing many different vehicle concepts, which means that the selected option must be flexible. Thus, the ability of each of the options to represent many unique concepts will translate into the metric of flexibility.

The second metric also stems from the requirement that the method must evaluate many different vehicle concepts. Due to the large number of concepts present in a typical architecture trade space, a manual process for creating the system level reliability estimates is undesirable. Therefore, the chosen option must be able to be automatically generated for each vehicle concept in order to maintain a practical evaluation time for the overall reliability assessment method. This will translate into the metric of automatic generation.

The first and simplest identified option for research question 4 is to multiply the subsystem level reliabilities to produce the system level reliability. The inherent assumption with this technique is that all the subsystems are in series. This means that any failure of a subsystem results in a failure of the system as a whole. Although this assumption may hold for some vehicle concepts, it does not allow for the inclusion of redundant systems. Therefore, this option is not flexible enough to evaluate a wide variety of vehicle concepts.

For the second metric, automatic generation, this option performs the best out of all four. Due to the simplicity of this option, no additional setup is required to produce the system level estimate. A single evaluation of a simple equation is all that is needed. Even though the simple multiplication method would be the most rapid, its lack of flexibility eliminates it from consideration.

The second identified option is the use of reliability block diagrams (RBD). In RBD, the subsystems can be represented using blocks that are connected based upon a physical decomposition of the system. Therefore, the RBD technique is very flexible because blocks can be set up in many different configurations. As was discussed in Section 3.2.1.6, series, parallel, k-out-of-n, switches, and other configurations can be included in the system model. For this reason RBD scores well for the metric of flexibility.

In terms of automatic generation, RBD is conducive to the idea. It is very possible to automatically generate a system block diagram based upon a set of rules and general system information. However, for complex systems this may become difficult, especially if switches and k-out-of-n type configurations exist. This leaves RBD as desirable in the area of automatic generation as long as the system diagram is not overly complex.

The third option for research question 4 was to use fault tree analysis (FTA), which was discussed in detail in Section 3.2.1.5. Similar to RBD, FTA utilizes a decomposition of the system as a graphical representation. In a sense, FTA can be considered an analog to RBD in that it is failure oriented while RBD is success oriented. For the same system, FTA and RBD should produce the same reliability predictions [171]. Thus, FTA possesses the same amount of flexibility as RBD because it is able to incorporate complex system configurations.

In terms of automatic generation, FTA is very similar to RBD. Many software tools exist for evaluating fault trees, and there are many potential approaches for automatic generation of the trees. Due to the analogous nature of FTA and RBD, both of these methods score the same against the two metrics of interest.

The fourth option for research question 4 is Stochastic Petri Nets (SPN), which was discussed in detail in Section 3.2.1.9. Stochastic Petri Nets utilize a local state-space representation of a system to evaluate the probability that the system will enter an undesirable state over a given period of time. An SPN model is very flexible and can contain many different system states and transition gates based upon multiple different probability distributions. In addition, the SPN is able to capture dynamic failure rates and aging effects. These abilities lead to the conclusion that SPN is the most flexible of all the identified options for research question 4.

Although SPN is considered to be the most flexible option, it does not fare well when considering the second metric, automatic generation. Due to the vast flexibility of the model, a large amount of information is needed to set up the SPN. This information includes the failure rates for each subsystem being modeled as well as the various states the subsystems may enter during operation. Due to the state-space representation used in SPN, each subsystem in the overall system would need to be modeled as a token. Using this approach introduces additional complexity when determining the proper links between the states and transitions. The appropriate rules for the passing of each subsystem between states would need to be incorporated in the model. Considering the details that need to be incorporated in the model, SPN is currently not conducive to automatic generation. For this reason SPN can be eliminated from consideration.

The final option for research question 4 is Markov chain analysis, which was discussed in detail in Section 3.2.1.8. Markov chains are global state-space representations of a system. Similar to SPN, Markov chains utilize states and transitions to model a system. However, Markov chains do not include the ability to use dynamic failure rates. In addition, Markov chains are limited to the use of exponential failure distributions [88]. These considerations make Markov chains much less flexible than SPN, but they can still be considered similar to FTA and RBD in terms of flexibility.

For automatic generation, Markov chains can be considered slightly better than SPN. However, the automatic generation of Markov chains is considered to be impractical. As was discussed in Section 3.2.1.8, Markov chains can suffer from a state-space explosion when modeling complex systems. If the system is made up of many different components, the number of states that are represented in the Markov chain can become prohibitively large. If the number of states is large, automatic generation of the model becomes very difficult and may require an unrealistic amount of computation time. As with SPN above, Markov chains will be eliminated from consideration due to their limitations in the area of automatic generation.

After considering the five options for research question 4, the FTA and RBD techniques can be identified as the most promising. Due to their similarity it is likely that either technique will be useful for application in the CONTRAST method. However, fault trees do hold a slight advantage. This is due to their failure oriented nature, which is in line with the Hall reliability growth method that uses number of failure modes as an input. For this reason, hypothesis 4 was derived, which identified FTA as the desired technique.

Hypothesis 4

If a system level fault tree is used to combine subsystem reliability estimates, then the resulting system level estimate will be more flexible than simple multiplication without encountering issues with automatic generation as with SPN or Markov analyses.

3.7 Experiment 3

Hypothesis 4 was developed in the previous section based upon an assessment of the perceived flexibility of the options for research question 4. At the conclusion of the section only two options remained for consideration, FTA and RBD. These remaining options will need to be compared across the second criterion, automatic generation. Since these techniques are very similar to one another it is expected that they both possess the same amount of flexibility. Therefore, the selection of one technique will come down to the ease in which the model can be automatically generated.

In order to assess this criterion, a literature search will be used to identify automatic generation techniques and software for RBD and FTA. This search will give an idea of how commonly these analyses are automatically generated. After conducting the literature search, it may be necessary to test the automatic generation of both RBD and FTA. This test will be conducted if the literature search yields very similar results for both analyses. The results of the literature search will be considered similar if the analyses have multiple approaches or commercially available programs for automatic generation.

If a test of automatic generation of RBD and FTA analyses is necessary, two additional criteria are proposed. These criteria are the setup and runtime of the automatic generation process. It is expected that for the same system, both techniques will produce the same reliability output. Thus, the setup and runtime of each technique is the deciding factor for which one will be applied in the CONTRAST method. To test the setup and runtime, automatic generation techniques or software will be selected based upon the literature search. An example matrix of alternatives (MOA) will be setup, which will contain representative vehicles. This MOA will allow for the evaluation of the runtime of the automatic generation of RBD and FTA across multiple different vehicle architectures. Ultimately, the time required will determine which technique will be applied.

If the setup and runtime are nearly the same for both RBD and FTA, the selection of an option will be considered inconsequential to the success of the CONTRAST method. If no difference in time exists between the options, then the method should be insensitive to this selection. Ultimately, hypothesis 4 will be substantiated if FTA performs better than RBD for the setup and runtime criteria. If the evaluation times are nearly identical, then the hypothesis will also be accepted. If RBD performs better than FTA, then the hypothesis will be rejected. Upon rejection, a new hypothesis will be formulated to reflect the results of Experiment 3.

3.7.1 Experimental Setup

3.7.1.1 Current FTA and RBD Tools

To begin the setup of Experiment 3, current tools for generating FTA and RBD analyses must be considered. Many current techniques exist for generating FTAs and RBDs, one of which may be suitable for application in the CONTRAST method. Table 9 displays the list of existing FTA and RBD tools that were reviewed for Experiment 3. In addition to the tool name and developer, columns were added to identify if the tool performed both FTA and RBD. The last column identifies the tools that are completely free of charge, free via a limited trial, or not available except by purchase. A website reference is given for each tool that offers a free download or trial.

Table 9: List of existing FTA and RBD tools

Tool	Developer	FTA	RBD	Free
RAPTOR [162]	USAF	Yes	Yes	Yes
Reliability Workbench	Isograph	Yes	Yes	No
Synthesis [136]	ReliaSoft	Yes	Yes	Trial
CAFTA [50]	EPRI	Yes	No	Trial
OpenFTA [5]	Auvation	Yes	No	Yes
RAM Commander	ALD	Yes	Yes	No
FTA Toolkit	Item	Yes	No	No

The list of tools in Table 9 is not meant to be an exhaustive list of tools available for use today. Many more tools for FTA and RBD exist, however, these tools all offer nearly identical capabilities. The goal of the list was to illustrate the various tool options available to the reliability analyst. The tools listed in Table 9 range from very complex integrated environments to simple GUIs. For example, the ReliaSoft Synthesis reliability software is meant to be an environment that can be used to integrate all types of reliability analyses including FMEA, RBD, FTA, FRACAS, maintenance analyses, and reliability growth. The simplest tool in the list is RAPTOR, which is a GUI based RBD or FTA program.

All of the tools in the list were initially considered for use in the CONTRAST method. However, after identifying that some of the software was not openly available, the list was pared down. The tools such as the Isograph Reliability Workbench and RAM Commander were eliminated from consideration because they were not available for use without purchase. The remaining options included two tools that are completely free, RAPTOR and OpenFTA, and two tools that allow for free trials, Synthesis and CAFTA.

The Synthesis reliability tool is the most complex of the tools in the list. This software is meant to be a platform for all other reliability analyses to be run and integrated. Although a free trial version of the tool was offered, it was not considered as a realistic option for application in the CONTRAST method. This is primarily due to the expected learning curve and the execution time of the analysis. Since the method will be using very simple FTAs or RBDs a complex platform such as Synthesis is not needed.

On the opposite side of the spectrum is the RAPTOR tool, which was originally developed by the U.S. Air Force. This tool consists of a very simple GUI that allows the user to assemble and calculate reliability block diagrams and fault trees. The tool is very easy to use and gives standard visualizations for both FTA and RBD. This simplicity makes RAPTOR desirable for application in the CONTRAST approach. However, the tool was only operable via the GUI, which would make automation of the FTA or RBD generation problematic. In order to run cases and assemble FTAs or RBDs automatically a wrapper would need to be created to interact with the GUI. It is expected that this operation would cause a very large increase in the required run time of the FTA or RBD analysis. For this reason the RAPTOR tool was also eliminated from consideration.

Considering the automation of the RAPTOR tool brings up a very important point when considering the pre-existing FTA and RBD tools. Due to the fact that the CONTRAST method is using very simple FTA or RBD, implementing an existing tool may cost more run time than it's worth. The existing tools all offer great options for visualization of the diagrams. They also allow for either quantitative or qualitative calculation of the FTA or RBD. Since the method will be evaluating thousands of vehicles in a single study the visualization options become less of a concern. There is not a vital need for visualizing each and every combination of FTA or RBD within a given matrix of alternatives. For this reason, the exploration of previous tools for use in the CONTRAST approach was terminated.

3.7.1.2 Automatic Generation of FTA and RBD

After eliminating the previous FTA and RBD tools from consideration an approach was needed for automatically generating the analyses. A custom approach was pursued because only simple FTA and RBD analyses will be required. This approach was deemed appropriate because the FTA and RBD equations can be operated on directly for such simple analyses. Therefore, the simple FTAs and RBDs within the CONTRAST method can be handled by assembling their equations.

In order to automatically assemble FTA and RBD equations an object oriented approach was taken based upon a defined matrix of alternatives. As discussed in Section 3.8 a morphological matrix will be used to define the vehicle concepts to be analyzed by the method. From this matrix an FTA or RBD equation needs to be generated for the vehicle architecture that has been selected.

The object oriented approach utilized two classes defined in the Python coding language. The two classes allow for the definition of each matrix row as an object and each specific component as an object. The class structures used in the CONTRAST method can be found in Appendix A.

In order to generate the FTA and RBD equations, the following process is taken within the Python code. First, each row of the defined matrix of alternatives is defined as a MOArow object. These objects possess attributes including the row name, number of options, list of options, row type, relationships, and dependencies. There are three row types utilized in the MOArow class; relation, numeric, and component select.

The relation type represents a matrix row that will change the parameters used by the FTA and RBD logic gates. For example, a matrix row for engine out may give an option such as n - 1 out of n. If this option is selected, FTA and RBD equations must be set up to be 2 out of 3 or 3 out of 4 depending upon the number of engines selected. Therefore, the relation type essentially defines the relationship between identical components.

The next row type is the numeric type. This row type is very simple and defines how many of a certain component will be used in the vehicle. An example of this type is the number of engines row, which may contain multiple options from 2 to 5. The numeric type defines how many instances of the specific component will be included in the FTA or RBD equation.

The final row type is the component select type. This type simply identifies the specific component to be used in the system. An example of this type is an engine selection row with options representing specific existing engines such as the SSME or RL-10. Following along with the previous example of number of engines, this row type will define the specific component object to be used in the FTA or RBD equation.

The next step in the code is to define all of the specific components within the matrix as component objects. The component class structure can be seen in Appendix A. Multiple different methods are defined within this class, which will be utilized later when performing the reliability growth calculations. Attributes within this class are defined in order to support the generation of the assumptions for the growth models.

After defining the component and row objects along with their dependencies an Entry function is called, which generates the FTA and RBD equations. This function assumes that each component will have its own "entry" within the FTA and RBD equation. In other words, the top level event in an FTA is connected to a number of entries that is equal to the number of components within the defined vehicle. Depending upon the relation and numeric rows defined in the matrix for the given component, the equation within the entry is adjusted accordingly. For example, if the SSME is chosen within the matrix of alternatives an entry for the SSME component will be created. If number of engines is selected as three and no engine out is selected, the entry for the SSME component will be generated as the cube of the individual engine reliability.

Following the generation of the entries within the FTA and RBD equations the overall reliability can be evaluated. This is carried out by assigning the reliability values of each component to their respective entries within the overall equation. The equation can then be evaluated, giving the system reliability.

3.7.1.3 Selection of FTA or RBD

After implementing an automatic generation code for both FTA and RBD, a test is needed in order to select a technique for application within the method. As discussed in Section 3.6 the chosen technique needs to be flexible enough to handle many different launch vehicle concepts, while allowing for rapid evaluation of each concept. Due to the similarity between FTA and RBD, the flexibility criterion is of little concern. Each of the techniques can easily represent very complex system diagrams. Therefore they are considered to be exactly the same in terms of flexibility.

The only remaining metric from Section 3.6 is the evaluation time of both techniques. This will require a test of the FTA and RBD generation code discussed above, which will identify any differences between the two techniques. In order to carry out this test a representative matrix of alternatives is required. This matrix will give the generation code a variety of vehicles in order to test the automatic generation capabilities of both FTA and RBD.

Table 10 shows the matrix of alternatives that was generated for Experiment 3. This matrix contains options that are representative of a typical launch vehicle morphological matrix. The representative matrix of alternatives also includes options for redundancy and engine out, which will test the ability of the generation code to put together more complex FTA and RBD equations.

Table 10: Representative launch vehicle matrix of alternatives

Avionics Redundancy	Yes	No		
Power Redundancy	Yes	No		
Number of Boosters	0	2	4	
Number of Engines	2	3	4	5
Engine Type	Engine 1	Engine 2		
Engine Out	None	n-2 out of n	n-1 out of n	

As can be seen from Table 10 a launch vehicle concept generated by this matrix will contain four components. These components include avionics, power, engine, and booster with the engine component allowing for a choice between two options. In an actual study these engine options may be specific existing engines, such as the SSME or RL-10 or generic engines representing a new development program. The redundancy and engine out options in the matrix represent relational row types, which will determine the structure of the FTA and RBD equation entries for the avionics, power, and engine components.

For definition of the component objects, basic assumptions regarding the reliabilities of the components will be used. These assumptions are not critical to the testing being carried out because this experiment is more interested in the generation of the FTA and RBD equations than the value of the output. The value of the output will only be considered if a major difference is seen between the output of the FTA and RBD equations.

In addition to the basic component reliability, a common cause failure (CCF) probability will be included in the assumptions. This probability will be introduced when redundancy or engine out is selected. The common cause failure probability will represent the case in which all of the redundant components fail simultaneously. The assumptions used in Experiment 3 for the component reliabilities and common cause failure probabilities can be seen in Table 11.

Table 11: Component reliability assumptions for Experiment 3

Component	Reliability	CCF Probability
Avionics	0.999	0.0005
Power	0.999	0.0005
Engine 1	0.0998	0.001
Engine 2	0.0996	0.002
Booster	0.99	0.0001

To complete Experiment 3 all of the possible combinations of vehicles from Table 10 will be selected and an FTA and RBD will be generated for each. Using the assumptions from Table 11 the FTA and RBD equations will be calculated for each vehicle concept and the result will be tabulated.

It is expected that the required run time for both of the techniques will be nearly identical because of the simplicity of the equations. Another expectation for Experiment 3 is that there will be little to no difference in the reliability outputs from FTA and RBD. This is expected because the techniques are very similar to one another. The only difference between the two techniques is that FTA is failure oriented while RBD is success oriented.

If the above expectations hold hypothesis 4 cannot be accepted or rejected based upon this experiment. This is because Experiment 3 will show that there is no difference between FTA and RBD, which means that the selection of a technique is a matter of preference. If the expectations do not hold and there is a difference between FTA and RBD in terms of run time or output, then hypothesis 4 can be accepted or rejected based upon the experiment outcome.

3.7.2 Experiment 3 Results

Using the class definitions given in Appendix A and the assumptions in Table 11, Python script was run in order to generate the FTA and RBD equations for all of the vehicles in Table 10. In total, 288 vehicle architectures were run, which required only 0.062 seconds of runtime. This runtime was divided evenly between the FTA and RBD generation scripts. Table 12 lists the run times of the script when generating both FTA and RBD equations simultaneously, only FTA equations, and only RBD equations. As expected the runtime for the FTA and RBD generation code is exactly equal.

Table 12: Runtime required to generate 288 reliability equations

Generation	Runtime (s)
FTA and RBD	0.0619
FTA only	0.03095
RBD only	0.03095

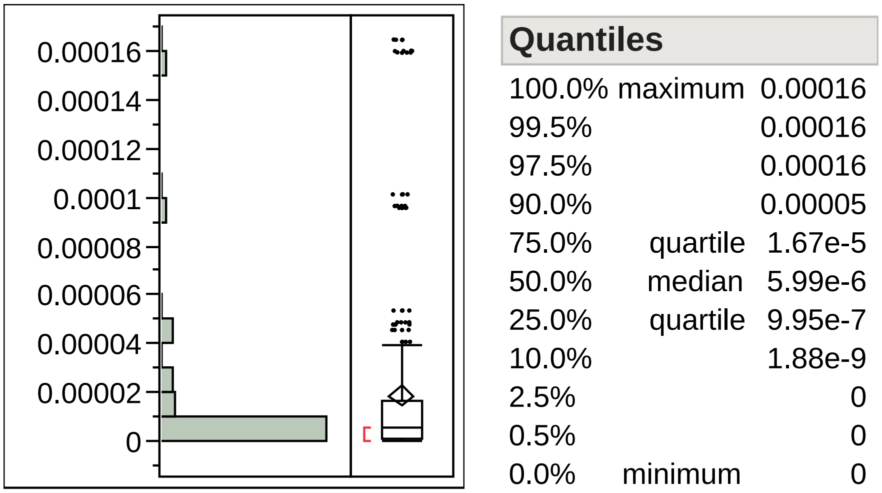
Since the runtime is exactly equal, the output of the analyses will be considered to see if there is any appreciable difference between FTA and RBD. Figure 27 below shows a distribution of the absolute value of the differences between the FTA output and the RBD output for all 288 vehicle architectures. It is interesting to note that 12 of these cases present a difference of 0.00016. This difference is not large, but because we are considering reliability a difference in the fourth digit is worth investigation.

After further investigation, all 12 of the identified architectures were found to contain 5 engines with no engine out capability. In calculating the reliability from the engine entry within the FTA and RBD equations it can be shown that the difference seen in Figure 27 is due to rounding error. The FTA and RBD equations for these cases can be seen below, with the engine reliability equal to 0.996.

$$R_{FTA} = 1 - (1 - R_{Engine}) * N_{Engines} \tag{24}$$

$$R_{RBD} = R_{Engine}^{N_{Engines}} \tag{25}$$

After plugging in the engine reliability as 0.996, the FTA equation gives a reliability of 0.98 while the RBD equation results in a non-rounded number of 0.980159. This ultimately explains the difference of 0.00016 that can be seen in Figure 27.



Quantiles		
100.0%	maximum	0.00016
99.5%		0.00016
97.5%		0.00016
90.0%		0.00005
75.0%	quartile	1.67e-5
50.0%	median	5.99e-6
25.0%	quartile	9.95e-7
10.0%		1.88e-9
2.5%		0
0.5%		0
0.0%	minimum	0

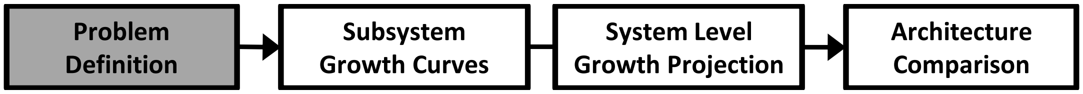
Figure 27: Difference between FTA and RBD reliability predictions

In Section 3.7.1.3 the expectations for Experiment 3 were discussed. These expectations were that the FTA and RBD approaches would be effectively identical after implementing the object oriented approach for generating the reliability equations. Table 12 and Figure 27 confirmed these expectations, showing that the runtime and reliability output of both the approaches are identical. The only exception is the identification of slight differences in the reliability output due to rounding errors within the code.

Due to the fact that the techniques cannot be differentiated, hypothesis 4 cannot be fully accepted or fully rejected. Ultimately the decision between FTA and RBD comes down to preference. Both of the methods will produce the same reliability output for the same vehicle architecture. The FTA method will therefore be chosen for application in the CONTRAST method.

The primary reason for this selection is based on the reliability growth model that was selected following Experiment 1. The growth model to be used in the method operates with assumptions pertaining to the number of failure modes and probability of occurrence of each of the modes. These assumptions fall directly in line with the failure oriented FTA approach. Within the FTA each entry can be considered a "failure mode" with the probability of failure equal to the probability of occurrence from the growth model. This alignment will be beneficial for the future application of the CONTRAST method.

3.8 Vehicle Architecture Definition



The completion of Experiment 3 finalized the definition of the analysis steps within the generic reliability and safety based decision-making process. Only one step remains to be addressed, Problem Definition. Recall that this step represents multiple different sub-tasks, which include establishing the need, establishing value, and defining the alternatives. The Problem Definition step will therefore require the definition of the metric of interest (LOM, LOV, LOC) as well as the general trade space to be explored. Within this trade space the specific vehicle options need to be identified, which will be fed into the reliability and safety analysis. Since the general trade space and metric of interest definitions will vary from project to project, only the technique for generating vehicle architectures for analysis will be considered.

There are two primary options for defining these vehicle architectures. The first option is the trade tree, which is a graphical decomposition technique. The trade tree decomposes the system either functionally or physically into a tree of parameters that are analyzed sequentially [116]. Due to the sequential analysis involved with trade trees, care must be taken in setting up the tree. As decisions are made going down the tree, other branches will be "pruned" or eliminated from consideration [116]. For this reason, the order of decisions being made is very important to the successful application of the trade tree technique. The setup time needed for a trade tree is thus much greater than other techniques. It requires a well thought out process to produce the order as well as all of the individual branches in the tree.

The second option for architecture definition is morphological analysis. Morphological analysis was developed in the 1940's as a method for assessing patterns in an orderly manner [179]. It utilizes a matrix of alternatives (MOA) to represent the physical or functional decomposition of the system [168]. In this technique the order of alternative selection does not matter, which means that any component of the architecture can be selected first. This benefit allows the analyst to avoid unintentionally eliminating options that may be desirable, which is a possibility when utilizing trade trees.

Morphological analysis also benefits from recent improvements to the MOA. One such example is the Interactive Reconfigurable Matrix of Alternatives (IRMA). An IRMA is an advanced MOA that includes compatibility information in the primary options table [168]. As the analyst makes architecture selections, the remaining options are highlighted based upon any incompatibility with the currently selected options. This example shows the additional capability of the MOA to store design metadata, which can be called upon when an option is selected. The ability to store such data may hold an added bonus for facilitating automatic generation of vehicle fault trees.

Morphological analysis was chosen for application in the CONTRAST method after a comparison of the two architecture definition techniques. Morphological analysis offers many more benefits than trade trees; however, it is important to note that either of the methods could be used for the initial vehicle definition. The selection of the technique is a matter of preference that will not affect the output of the method. Architecture definition was not posed as a research question in this case because it was considered to be beyond the scope of the research objective. A majority of this research will address the assessment of the expected reliability of a vehicle concept, which is independent of how the concept was initially generated.

3.9 Research Question 5: Reliability Growth Model Assumptions

The selection of a matrix of alternatives approach for generating vehicle architectures concluded the first pass through the generic steps of the reliability and safety based decision-making process. At this point the primary components of the CONTRAST method have been identified. However, additional research questions are required to address specific details for method application. The first of the additional research questions addresses the production of the assumptions for the reliability growth model.

Research Question 5

How can the reliability growth assumptions be produced at the subsystem level?

The primary motivation behind research question 5 is the growth model accuracy, which is directly affected by the traceability of the assumptions. If the model assumptions are not traceable, the output of the method will also be easily discreditable. In order to address research question 5, the assumptions required by the growth model identified in Section 3.3.2.3 will be considered first.

The Hall growth model contains three primary assumptions; number of failure modes, probability of occurrence, and fix effectiveness factor. The first assumption, number of failure modes, is simply the number of events that can cause the top level event of interest such as LOC or LOM. The second assumption provides the probability of occurrence for each of these events, which is typically defined using some form of continuous distribution. As discussed in Section 3.3.1.5, Hall suggests a Beta distribution to define the probabilities of occurrence. The third assumption, fix effectiveness, accounts for potential design fixes that will be implemented throughout the vehicle's life-cycle. This parameter represents the reduction in probability of occurrence for a given failure mode if that mode were to occur. A value of 1 represents the complete elimination of the mode from the system, while a value of 0 means that no fix was implemented.

After considering the required reliability growth assumptions, multiple approaches for generating them can be identified. Three primary options exist for generating the assumptions; subject matter expert input, parts count method, and failure mode and effect analysis. The latter two options are pre-existing reliability methods that were discussed in Sections 3.2.1.2 and 3.2.1.3. Comparison to previous systems could be considered an additional option for producing the growth assumptions. However, all three of the approaches will rely upon historical data of some sort. For this reason, comparison to previous systems was not identified as its own approach for generating the reliability growth assumptions.

The first option for producing the growth model assumptions is to use subject matter expert input. This option equates to an "ask the expert" approach, which involves setting the assumptions based upon engineering judgment. To apply the SME input option, a SME in each subsystem area would need to be identified. Based upon prior knowledge and experience the SME in each field would then generate the growth assumptions for each subsystem under consideration.

In terms of traceability of the assumptions, this option is considered to be behind FMEA and parts count. The primary reason for this is that the SME input process is much less structured. While gathering the SME input there is no guarantee that the SME's polled are qualified to make the assumptions for each subsystem. In addition, the reasoning or previous knowledge used by the SME to generate an assumption is not captured. Therefore, the traceability of the assumptions can typically boil down to ”Expert A said it, therefore it must be true”, which is undesirable.

The SME input option is less traceable but it is not necessarily the worst option for all of the reliability growth assumptions. For the first two assumptions, which are easily quantifiable using the right data, this approach is the worst in terms of traceability. However, the third assumption is set up fairly well for an SME based approach. This is because in practice, the fix effectiveness factor is very difficult to quantify.

As stated earlier, the fix effectiveness factors represent the amount of reduction seen in the probability of occurrence of a given failure mode. In order to quantify these values failure modes for a system would need to be tracked both before and after fix attempts are made during the vehicle life-cycle. This implies that the probability of occurrence of the mode can be calculated prior to the fix and the new probability of occurrence can be observed after the fix. The fix effectiveness factors will also be affected by non-quantifiable parameters such as the rigor of the test and redesign process or the specific failure reporting and correction methods the program will employ. The SME input method would be well suited to predict the fix effectiveness factors for this reason. The SME's will be able to apply their knowledge of the test-fix-test process to help quantify the fix effectiveness.

The next option for reliability growth assumption generation is the parts count method, which was discussed in more detail in Section 3.2.1.2. This method begins with identification of the individual parts that make up the system being analyzed. In other words, considering the levels of characterization from Section 3.3.2.2, PCM counts the number of items in the level just below the level of analysis. For example, if the PCM approach is being used for a subsystem, all of the major assemblies will need to be identified. To determine the system reliability, each identified part is assigned a reliability value. Typically these values are simply multiplied across all the parts in order to estimate the system reliability.

The parts count method is particularly well suited for the number of failure modes assumption in the Hall model. Carrying out the parts count method at a given level of characterization will result in a list that is analogous to the number of failure modes at that level. The number of failure modes at a given level can be equated to the number of events at the next lowest level in an event tree. It is expected that the failure of each part counted by PCM would show up in that event tree. This is especially true when considering top level events such as LOC. An LOC event can usually be tied to a catastrophic failure of some sort. Each of the parts identified using PCM will most likely have at least 1 catastrophic failure mode, resulting in an equivalent number of events linked to the top level.

For the second and third assumptions the parts count method is less desirable. Although PCM does account for the probability of occurrence through the assigned reliability values for each part, it can be considered nearly the same as the SME input option. The reliability values for the parts in PCM can be generated using analysis, but they are typically generated using comparisons to previous systems. Another approach for assigning reliability values in PCM is simply to "ask the expert", which is the same as SME input.

The final option for assumption generation is the use of an existing reliability technique, failure mode and effect analysis. The FMEA approach was discussed in detail in Section 3.2.1.3 and involves identifying and tabulating all failure modes within a given system along with their effects and criticality. The FMEA worksheets are therefore well suited to producing all three of the reliability growth model assumptions.

The first growth assumption is addressed directly by FMEA because its goal is to produce a comprehensive list of failure modes for the given system. This assumption can be made very easily and traceably utilizing an FMEA worksheet. However, it is important to note that the FMEA worksheets contain more detail than is required. Since the goal of FMEA is to identify all potential failure modes, there will be many modes within the full worksheet that do not directly map to the top level event of interest. This is especially true in systems that utilize redundancy or fault detection and correction. Both of these techniques will reduce the number of part or component failure modes that will lead directly to the top event. If FMEA is to be utilized for production of the reliability growth assumptions, some simplifications of the worksheet may be required. This is due to the amount of time required to produce a comprehensive FMEA worksheet for a complex system.

The next assumption to consider is the probability of occurrence of the failure modes. This assumption is addressed by FMEA if the worksheet includes criticality, which is referred to as FMECA or failure mode, effect, and criticality analysis. The criticality values are typically split into different categories from minor to catastrophic [40]. In addition, the failure modes can be assigned a probability of occurrence in either qualitative or quantitative form. The qualitative probability of occurrence scale assigns the modes to one of five levels ranging from extremely unlikely to frequent [40]. The quantitative probability of occurrence is appropriate only when a source is available for failure rate data. Even if the qualitative scale is used, the probability of occurrence for each mode could be estimated using the FMECA worksheet.

The third growth assumption, fix effectiveness factor, is also indirectly addressed within FMEA. As shown in Figure 10 in Section 3.2.1.3, the FMEA worksheet contains columns for failure detection methods and compensating provisions. The first of these columns simply captures the techniques that are applicable for detecting the given failure mode. The second column allows for the identification of options for preventing or reducing the probability of occurrence for the failure modes. These columns could be used to qualitatively assign values for the fix effectiveness factors. If a given failure mode has many detection methods as well as many options for prevention, its fix effectiveness factor will likely be high. On the other hand, if a given mode has no detection or prevention options its fix effectiveness factor will be much lower. The only issue with utilizing FMEA for qualitatively assessing the fix effectiveness factors is the amount of time required to fill in these columns. For all of the failure modes in the worksheet, options for detection and prevention or correction must be explored, which can be very time consuming for a complex system.

After considering all three options for generating the reliability growth model assumptions, a few key observations can be made. The first of these observations is that none of the identified options are a clear cut choice to produce all three of the growth model assumptions. The SME input option seemed to be applicable only to the third assumption, while the PCM and FMEA options were better suited for the first and second assumptions.

The second observation is in regard to the benefits of using FMEA. First, FMEA can be considered the most traceable of the identified options. This is because FMEA is a very standardized and structured process, where as SME input and PCM are not. The FMEA process is also expected to benefit the most from historical data stemming from the design work of previous programs. The FMEA worksheets can be “living” documents that are iterated upon as the program progresses through the design process. This means that FMEA data from previous systems may exist even if the program was canceled prior to operations.

Based upon these observations, two use cases can be identified to address research question 5. The first is the case where FMEA data is available from a previous system that is deemed similar to the new system. In this case, the FMEA option is the most appropriate for generating the number of failure modes assumption as well as the probability of occurrence assumption. Both of these assumptions can be made based upon previous data, which is considered more traceable than the other options. As discussed above, the SME input approach is the most well suited for the third assumption, fix effectiveness factors. Due to the fact that this assumption is difficult to quantify in practice, it has been deemed acceptable for SME input to generate the fix effectiveness factors.

The second case for research question 5 occurs when FMEA data from a previous system is not available for comparison. In this case a full FMEA for the new system may require a significant time commitment, which will be impractical when considering a large architecture space. As mentioned above, the PCM approach can serve as a very quick method for identifying the key failure modes within the system. Therefore, in the case where previous FMEA data is not available the PCM approach can be used to generate the initial list of failure modes.

In the absence of reliability data, the probability of occurrence values for the failure modes from PCM must be assumed in a generic fashion. The literature regarding the development of the reliability growth model selected in Section 3.3 gives suggestions regarding the probability of occurrence distributions. For the second case, the generic assumptions from the literature will be applied in lieu of actual data. This approach will ultimately be tested during the example problem in Section 4.2 to ensure its validity. The final assumption for the second case can then be determined using the SME input approach.

The traceability of the assumptions for the growth model is of utmost importance to the accuracy of the resulting reliability estimates. A FMEA approach was identified as the most traceable method for producing the growth assumptions, however, it has shortcomings in terms of time required to complete the analysis. With traceability in mind, the FMEA approach was chosen as the backbone for the production of the growth assumptions, which will be augmented with two other techniques. The two cases identified above for research question 5 have been developed to provide rapid yet traceable assumptions for the growth model chosen in Section 3.3. This will preserve the defensibility and the accuracy of the resulting reliability estimates. From the two cases above a two part assertion to research question 5 was developed.

Assertion to Research Question 5

- If detailed data from a previous, similar system exists, then the number of failure modes and probability of occurrence assumptions can be generated based upon this data with the fix effectiveness factors coming from SME input.

- If detailed data from a previous, similar system does not exist, then the number of failure modes assumption can be rapidly estimated using the PCM approach and the probability of occurrence distribution must be assumed to represent a generic complex system. The fix effectiveness factors can be generated using SME input.

3.10 Research Question 6: Application of Growth Curves to a Fault Tree

The second additional research question is in regard to the application of subsystem growth curves to the system level fault tree. As discussed in Section 3.6 the growth model will be used at the subsystem level and the system level reliability will be generated using FTA or RBD. There are multiple different approaches that can be used to apply the subsystem level growth curves to an FTA or RBD. Research question 6 was posed to address this issue for future applications of the method.

Research Question 6

What method is most appropriate for applying subsystem reliability growth curves to a system level fault tree?

To address research question 6, three candidate options can be identified. The first option is to anchor all of the subsystem reliability growth curves at equivalent flight 0. With this approach, each step in time at the system level equates to the same step in time at the subsystem level. For example, to calculate the system level reliability at equivalent flight 0, the reliability distributions for equivalent flight 0 for each subsystem will be used. Similarly, the system level reliability at equivalent flight 100 will be calculated using the reliability distributions for equivalent flight 100 for each subsystem.

The second option for research question 6 is to step through the individual subsystem reliability growth curves based upon assumed failures. At each step in time the probabilities of failure could be used to evaluate which subsystems operated successfully or failed. If a subsystem was said to be failed, its growth curve would be incremented by a single flight. This increment would represent the improvement in reliability after the failure was identified and corrected. To evaluate the system level growth using this option another MC would be required.

The final option is to anchor each subsystem growth curve at an initial point based upon an assumed test or flight schedule. With this option an assumed test schedule would be used to set the initial point for each subsystem. For example, liquid rocket engines may undergo 50 missions worth of hot fire testing, while a solid rocket booster may only undergo 15. In this case the system level reliability estimate at equivalent flight 0 would use the reliability distributions at flight 50 and 15 for the liquid engines and solid rocket booster, respectively.

This option can also be applied in the case where unique development programs are started at different points in time. For example, a liquid engine development program may be assumed to start at the initiation of the program, while development for an advanced solid booster may begin much later. In this case the growth curve for the liquid engine will be anchored at equivalent flight 0 while the growth curve for the advanced booster may be anchored at equivalent flight 50 or 100. Applying the second option in this manner will be of use when considering a block upgrade approach for vehicle development. The relevance of this approach for considering block upgrades will be discussed in more detail in Chapter 5.

To evaluate the three options identified above, a set of selection criteria is needed. Since research question 6 affects the process of the CONTRAST method it requires similar considerations to research question 5. First, the prediction accuracy of the chosen option is of utmost importance. In order for the method to achieve success, it must be able to produce accurate results. Therefore, accuracy is an obvious candidate as a selection criterion for research question 6.

Next, the evaluation time of the selected option is also an important consideration. The number of concepts being evaluated has a great effect on the overall run time of the method. Thus the option chosen here will need to have an acceptable evaluation time.

Finally, data availability is also applicable to research question 6. The third option will require estimates for test and flight schedule of the vehicle being analyzed, which implies the use of outside data. Therefore, data availability will be added as the third selection criterion.

After identifying three selection criteria, the options for research question 6 can be compared. The first option, anchoring all growth curves at flight 0, is the simplest of the three. This option requires no additional data or changes to the implementation of the fault tree analysis. Therefore, option one is considered the most desirable in terms of data availability as well as evaluation time. However, the accuracy of the output may be affected by using this option. For actual development programs it is very rare for all of the subsystem developments to occur simultaneously. This means that anchoring all of the growth curves at flight 0 may not be an accurate representation of reality.

The second option for research question 6 is expected to require the longest evaluation time of the three identified options. At each step in time option 2 requires a MC run to evaluate which specific subsystems were assumed to have failed. This requirement will greatly increase the run time of the method, especially when many increments in time are required. Similar to option 1, option 2 will benefit from avoiding the use of outside data. It therefore performs better than option 3 against the data availability criterion. However, the second option may have issues with prediction accuracy. Since this option would require many MC runs at each step in time, it is expected that the distributions produced by this approach would have very large variances. These distributions may have large enough ranges such that the comparisons between vehicles become meaningless. The second option is thus considered as the worst in terms of accuracy.

The final option requires the development of an estimated test or flight schedule for the vehicle being analyzed. The addition of schedule would require some outside data in order to complete the analysis. Therefore, data availability issues may arise when using this option. However, the use of an estimated test or flight schedule would relieve some of the inaccuracies seen in the first option. If a schedule could be produced, all of the subsystem growth curves could be shifted to their appropriate initial values. This would result in better prediction accuracy than option 1. However, the third option fares worse when considering the evaluation time criterion. Due to the requirement of an assumed test or flight schedule, the amount of time to set up this option increases. The generation of the schedule will require additional time for data collection, causing this option to have a longer evaluation time than option 1 and option 2.

In comparing the three options for research question 6 across the identified selection criteria, it was concluded that option 1 is the most appropriate. Option 1 is expected to perform the best in two of the three selection criteria. The prediction accuracy is a potential issue, but with careful setup of the analysis it is expected that any issues can be mitigated.

Hypothesis 6

If all subsystem growth curves are anchored at equivalent flight 0, the resulting system level growth curve can be produced without increasing evaluation time or encountering data availability issues while maintaining an acceptable prediction accuracy.

3.11 Experiment 4

Following the development of the options for research question 6, a test is required to either accept or reject hypothesis 6. This test will examine the application of each of the options in more detail. In order to facilitate the selection of the best option for application in the CONTRAST method, three criteria were derived in Section 3.10. These criteria are prediction accuracy, evaluation time, and data availability.

The first criterion for Experiment 4 is the prediction accuracy of the selected option. This criterion refers more to the capture of the desired reliability growth trends than the numerical result. The numerical results should be very similar since each of the options in Experiment 4 will utilize the same failure mode assumptions for the reliability growth models. The prediction accuracy criterion will be evaluated based upon the observed output of each option. The research objective is to produce a method that can project reliability of launch vehicles from program initiation to maturity. Therefore, reliability growth results that show the expected initial reliability of the vehicle at program inception along with the entire growth curve up to vehicle maturation are desired.

In addition to the qualitative assessment of the prediction accuracy in Experiment 4, each of the options will be applied during the example problem in Section 4.2. This will allow for a quantitative comparison of the errors for each option when compared to real launch vehicle data. This exercise will further support the conclusions regarding hypothesis 6, which will be developed in the results section for Experiment 4.

The second criterion for Experiment 4 is data availability, which will assess the amount of outside data that is required for setup of each of the options. For option 3 this data includes the assumed test or flight schedule, while for option 2 it represents additional information and setup required for time incrementing in the model. This criterion was selected in order to compare the data requirements of each of the options. Reliance upon less outside data is considered to be a positive in this case because there will be less opportunity for introducing errors into the model.

The final selection criterion, evaluation time, can be assessed simply by running the models for a set of vehicle architectures. This criterion was selected because of its importance to the future application of the CONTRAST method. When carrying out trade studies on very large architecture spaces the evaluation time of the model will limit the total number of vehicles that can be evaluated. This ultimately traces back to the original requirements that were laid out with the research objective.

To execute Experiment 4, the chosen growth model from experiment 2 will be setup first using the Python coding language. After setup of the growth model, an FTA will be generated for each vehicle being analyzed. Following the generation of the FTA, the reliability growth model will be run using each option for research question 6. The results of the model run will then be used to evaluate the accuracy and evaluation time criteria.

3.11.1 Experimental Setup

In order to test the three options for research question 6, a representative matrix of alternatives must be generated. This matrix will include many different notional launch vehicle architectures and will help test the required evaluation time of each option. For the entire set of vehicle architectures in the matrix each of the options for research question 6 will be used to generate a reliability growth estimate. These estimates can then be used to assess the evaluation criteria that were defined above.

The previous experiment developed a representative matrix of alternatives, which was presented in Table 10. This matrix can be reused for Experiment 4 because it contains a sufficient number of vehicle architectures to test each option. In all, 288 unique architectures can be generated from Table 10.

The reliability growth assumptions for each of the subsystems within the matrix of alternatives must be developed. The first subsystem to consider is the solid rocket booster. This booster will be assumed as similar to the solid rocket boosters that were used on the STS. For each booster a more detailed list of high level failure modes can be identified, which is shown in Table 13. From this generic list the number of failure modes assumption for the solid booster was set to 6.

Table 13: High level Space Shuttle SRB catastrophic failure modes

Mode	**Explanation**
Case joints	Rupture of the joints between segments
Thrust vector control	Failure to properly align thrust from each booster
Ignition	Failure of ignition system causing uncontrolled burn of propellant
Separation	Failure to separate from the external tank

The next subsystem to consider is the liquid rocket engine. In the representative matrix of alternatives two engine options are given in order to represent unique engine types. The first engine will be assumed to be a gas generator, which is a common cycle due to its relative simplicity [156]. The second engine will be assumed as a staged combustion cycle, which is more complex. This engine is assumed to be similar to the SSME.

The SSME is a staged combustion cycle engine, which utilizes dual turbopumps for both the fuel and oxidizer [133]. Key components in addition to the turbopumps include the fuel and oxidizer pre-burners, main combustion chamber, nozzle, and thrust vector hydraulics. Using this list, Table 14 was generated to identify the high level catastrophic failure modes for the SSME. Since engine 1 is assumed as a less complex option it will be assumed to contain only a single turbopump each for the fuel and oxidizer. This results in 7 failure modes for engine option 1.

Table 14: High level SSME catastrophic failure modes

Mode	**Explanation**
Fuel turbopump (2x)	Failure of the low or high pressure pumps to provide fuel flow
Oxidizer turbopump (2x)	Failure of the low or high pressure pumps to provide oxidizer flow
Fuel pre-burner	Failure to ignite or uncontrolled burn of propellant
Oxidizer pre-burner	Failure to ignite or uncontrolled burn of propellant
Combustion chamber	Uncontrolled or unstable burn of propellant
Thrust vector control	Failure to direct engine to desired angle
Nozzle	Loss of integrity of nozzle structure or failure of heat exchanger

The next vehicle subsystem to consider is the power subsystem. This subsystem may include key components such as fuel cells or batteries, auxiliary power units, power distribution, and power conditioning units [92]. For Experiment 4 a list of four primary modes were identified for the power subsystem. These modes can be seen in Table 15.

The final vehicle subsystem to consider is the avionics subsystem. Launch vehicle avionics subsystems can be very complex and typically include the flight computers (hardware and software), data handling and processing, and instrumentation (flight controls, sensors, etc.) [92]. For the generic vehicle setup for Experiment 4 a list of six high level failure modes have been identified for the avionics subsystem. Table 16 lists these modes.

Table 15: High level power subsystem catastrophic failure modes

Mode	**Explanation**
Power source	Failure of the primary power source to provide adequate power
Auxiliary source	Failure of secondary source to provide adequate power
Power conditioning	Failure to condition the output power to appropriate voltage for vehicle systems
Power distribution	Failure to distribute power to vehicle systems

Table 16: High level avionics subsystem catastrophic failure modes

Mode	**Explanation**
Flight computer(s) hardware	Physical failure of flight computer hardware
Flight computer(s) software	Failure of flight computer software causing loss of guidance and navigation
Data handling and processing	Failure to successfully process vehicle information leading to abort or vehicle destruct
Flight controls	Failure to provide required control inputs or total loss of vehicle control
Sensors	False indication of off nominal conditions leading to abort or vehicle destruct
Data distribution	Loss of linkages between flight computer and control actuators or other critical systems

After completing the number of failure modes assumptions, two growth model parameters remain. These parameters are the probability of occurrence of the failure modes and the fix effectiveness factors. First, the fix effectiveness factors will be set.

As discussed in Section 3.4 the fix effectiveness factors for Experiment 1 were set based upon values given in the literature from Hall and Morse. The fix effectiveness values will ultimately depend upon the details of the vehicle development program, which includes the implementation of rigorous post flight data analysis, failure reporting techniques, and failure prevention and review boards. Based upon what techniques are applied for the system the fix effectiveness factors are expected to change.

Due to the fact that Experiment 4 is modeling notional vehicles a single set of assumptions will be used for the fix effectiveness factors across all of the vehicles. This is based off of the assumption that each of the vehicle concepts would be developed by the same organization utilizing the same failure prevention and reporting techniques. The fix effectiveness factors for this experiment will therefore be modeled as uniform distributions ranging from 90 % to 99 %. Next, the probability of occurrence assumptions will be set for both the system and subsystem level.

The system level probabilities of occurrence were addressed previously during Experiment 1 in Section 3.4. In this section multiple different sets of Beta distribution parameters were given from the literature for the growth models. Since Experiment 4 is considering notional vehicles and the goal of the experiment is to compare output from vehicle to vehicle, a set of parameters will be selected that will be used for all vehicles. This will enable more direct comparisons between different vehicle concepts. The system level probability of occurrence assumption, therefore, will be based upon the same Beta distribution as in Experiment 1. This distribution has the parameters $\alpha = 0.22$ and $\beta = 8.75$. For the subsystem level, new parameters must be calculated based upon the system level distribution.

In order to calculate the subsystem level probability of occurrence assumptions the system level probability of occurrence along with the number of failure modes at the subsystem level must be utilized. This will be done in order to ensure that the assumed subsystem probabilities of failure will match the calculated probability of failure based upon the identified subsystem failure modes. For example, consider the power system assumptions from above. The system level growth model will assume that a power subsystem failure will occur with the probability of failure distributed as $Beta(0.22, 8.75)$. At the subsystem level, four primary modes were identified that lead to the power system failure. The probability of occurrence of these modes must be defined such that the resulting probability of power system failure is approximately distributed as $Beta(0.22, 8.75)$. Therefore, the following equation will hold for determining the distribution for the probabilities of occurrence at the subsystem level:

$$p_{system} = p_1 + p_2 + p_3 + ... + p_n \tag{26}$$

where, p_{system} is the probability of occurrence of the system failure distributed as $Beta(0.22, 8.75)$ and $p_1, p_2, ...p_n$ represent the probabilities of failure of the subsystem failure modes distributed as $Beta(\alpha, \beta)$. The parameters for the Beta distribution of the probabilities of occurrence for the subsystem failure modes can be solved for numerically. This was carried out by picking the parameters α and β and drawing n probabilities of failure from $Beta(\alpha, \beta)$ to give values for p_{system}. A fit was then produced to the values of p_{system}, which was compared to the original distribution $Beta(0.22, 8.75)$. The process was repeated until convergence upon this distribution was achieved. A more detailed explanation of the probability of occurrence distribution derivation is given in Appendix C. Table 17 contains a list of the final Beta parameters for each of the subsystem probabilities of occurrence.

Table 17: Reliability growth assumptions for Experiment 4

Component	# Modes	Beta Parameters
Avionics	6	0.0421, 10.04
Power	4	0.08, 12.1
Engine 1	7	0.046, 12.03
Engine 2	9	0.0313, 11.08
Booster	6	0.0421, 10.04

After identifying the vehicle architectures and component reliability assumptions for the subsystems, the assumptions for the development schedule are needed. These assumptions will be used in option 3, which anchors all of the subsystems based upon their assumed first flight. This means that the subsystems that begin development earlier in the vehicle life-cycle will progress through more testing and reliability growth than other subsystems that are developed later.

In order to develop these assumptions the history of the Space Transportation System can be used. The Space Shuttle will be used to produce representative assumptions because of its extensive flight history and availability of program documentation. The assumptions that are needed for option 3 are the assumed first flights for each of the components defined in Table 17 above. Assuming that the program begins at equivalent flight 0, the equivalent "first flights" of the components can be estimated based upon their initial testing dates. The STS program will be sufficient to provide these assumptions for the avionics, power, booster, and Engine 2 components. The assumption for Engine 1 will be developed assuming that the engine has been flown before and benefits from some flight heritage. The first flight for Engine 1 will therefore be set to equivalent flight 0 with the other components coming thereafter.

The remaining component assumptions can be developed using the STS program history. The SSME will be used as a surrogate for Engine 2, the SRB for the booster, and the orbiter for both the avionics and power subsystems. The estimation of the equivalent first flights of these components depends upon their timeline for developmental testing. Therefore, a timeline of key events from the STS program is required.

The STS program took shape during the early 1970's following many reusable launch vehicles studies tracing back to the mid-1960's [173]. The Space Shuttle was first formally endorsed by President Nixon in January 1972 [144]. Following this endorsement the definitive contracts for the orbiter, SSME, and solid rocket boosters were all signed later in 1972 [173]. The first of these components to be tested was the SSME, which ran a full ignition test in June 1975 [11]. Two years later in February of 1977, flight testing of the Space Shuttle orbiter began at Edwards Air Force Base [173]. The beginning of flight testing will be assumed to be equivalent to the beginning of the testing of the avionics and power subsystems contained within the orbiter. Following the start of orbiter flight testing the first full scale SRB test was completed by Thiokol in July 1977 [144]. Finally, the first flight of the STS vehicle was successfully completed four years later on April 12, 1981 [144].

As can be seen from the previous discussion, the timeline for the STS program from initiation to first flight was between January 1972 and April 1981. During these 9 years, developmental testing of the various components was completed. To generate the assumptions for option 3 a new timeline will be assumed based upon the STS timeline discussed above.

The beginning of the new timeline will start with the first equivalent flight of Engine 1, which is the assumed "heritage" engine in this case. Since Engine 1 is assumed to have flown prior to the initiation of the new program it will be anchored at equivalent flight 0. The beginning of the new program will then be assumed to occur at equivalent flight 50, giving the heritage engine 50 flights worth of reliability growth. The remaining components will be set based upon the STS timeline.

As discussed above the SSME was the first component to undergo full testing.

Therefore, the SSME first flight will be set to equivalent flight 50 on the new timeline. Details regarding the SSME test program can then be used to place the SRB and orbiter on the new timeline.

The SSME program was initiated two years prior to the start of testing on the orbiter and SRB [144]. During this program the engines underwent nearly 100 equivalent missions worth of hot fire testing prior to the first operational launch of STS [11]. This equates to around 20 equivalent missions worth of testing per year of the test program. With this number in mind a crude estimate for the beginning of the SRB and orbiter test programs can be obtained. Assuming 20 equivalent flights per year, the orbiter first flight will fall around 70 on the new timeline. Therefore, the avionics and power subsystems will be assumed to have a first flight of 70. Since the avionics system can be considered more "complex" than the power system the avionics will be assumed to have undergone more testing for the purpose of Experiment 4. Thus, the avionics first flight will be adjusted to 60 and the power system first flight will remain at 70.

The final component to consider is the solid rocket booster. As discussed above, the SRB testing began about 6 months after the beginning of the orbiter flight testing. Using the simple assumption of 20 flights worth of testing per year, the 6 month difference equates to approximately 10 additional equivalent flights. The SRB first flight will therefore be set to 80 on the new timeline.

In order to conclude the first flight assumptions, the new timeline must now be reversed in order to reflect the appropriate amount of development time for each component. The timeline will be reversed because the assumed first flight in option 3 is the equivalent flight at the subsystem level at which the first operational flight of the vehicle is assumed to occur. This means that the heritage engine will have quite a bit of development time prior to the first operational launch, meaning its first flight will be set to a high number. The high number for first flight represents the amount of testing or previous flight heritage the component has gained prior to the first operational flight of the new vehicle. Table 18 gives the updated first flight values that will be used in Experiment 4.

Table 18: Reliability growth assumptions for Experiment 4

Component	**First Flight**
Avionics	40
Power	30
Engine 1	100
Engine 2	50
Booster	20

A few additional assumptions are required to complete Experiment 4. These assumptions include the fix effectiveness factors, the number of repetitions, the total number of flights, and the number of steps in time. The fix effectiveness factor assumptions were addressed in Experiment 1, which can be leveraged for this experiment. Therefore, the fix effectiveness factors will be modeled as a uniform distribution between 95% and 99% for all of the subsystems.

The total number of flights and the number of steps in time will be chosen to ensure a long enough flight history without requiring an unmanageable number of runs. The total flight number was chosen to be 300 flights, which is between the total flight history of the Soyuz and STS vehicles from Experiment 1. In order to reduce the total runtime, 60 steps in time will be used, which means reliability results will be produced at every 5th flight throughout the history.

The final assumption for Experiment 4 is the number of repetitions. This assumption refers to the number of random draws that will be executed at each step in time in order to produce the system level reliability estimate. As with the previous assumptions, the number of repetitions is primarily constrained by the required runtime. In order to keep the runtime at a reasonable level, 1000 repetitions will be used. With this number of repetitions along with the 60 steps in time and 288 unique vehicles, 17,280,000 total runs will be completed using each option for Experiment 4.

After completing the assumptions for each of the options the expectations for Experiment 4 can be discussed. During the introduction of Experiment 4, three criteria were identified for testing the options. These criteria are prediction accuracy, data availability, and evaluation time. The numerical experiment that will be carried out using the assumptions developed above will test the prediction accuracy and evaluation time metrics. The data availability criterion will be assessed qualitatively. A discussion of this criterion is given in the results section for Experiment 4.

For the first criterion, prediction accuracy, the results from each option will be analyzed for all 288 vehicles simultaneously. The lumping of all vehicles will allow for easy identification of the general trends shown by each incrementing option. As discussed previously, the prediction accuracy criterion relates to the capture of the reliability growth of the system across the whole life-cycle. Therefore the results will be examined for trends that illustrate the expected initial reliability, mature reliability, and number of flights to reach maturity.

The expected result for this criterion is that option 1 will show a much more pronounced reliability growth curve than options 2 and 3. Option 1 anchors all of the subsystem reliability growth curves at equivalent flight 0, which should lead to a relatively well behaved system level growth curve. This option essentially guarantees that the output will provide all of the desired characteristics.

Option 2 anchors each of the subsystems at equivalent flight 0 but it uses an alternative incrementing scheme. In this approach, the subsystems are incremented based upon an assumed failure. For each step in time random draws are performed based upon the current reliability of each subsystem. If the draw determines that the subsystem "failed" the subsystem reliability growth curve will be incremented.

This approach is expected to introduce extra variability into the system level output. Ultimately this option is expected to produce a very wide range of expected system reliabilities that will not produce all of the desired characteristics.

In terms of prediction accuracy option 3 is also expected to perform worse than option 1. Option 3 anchors each of the subsystems at a non-zero equivalent flight, which represents the varying test schedules of the subsystems. Although this more closely mimics actual programs, option 3 does introduce extra requirements for input data. It will also produce an initial reliability that pertains to the first operational flight, which will be larger than the initial reliability value that is a desired output.

The third criterion, evaluation time, is expected to be very similar for all three of the options. Since option 2 introduces additional random draws for each case, it is expected that this option will have the longest total runtime. Option 3 introduces alternative first flights for each of the subsystems, which may also increase the required runtime. The first option is the simplest in this case and is therefore expected to require the least amount of runtime.

As stated in hypothesis 6, option 1 is expected to perform the best in terms of prediction accuracy, evaluation time, and data availability. In order to accept this hypothesis the following criteria will be used. First, the discussion of the data availability criterion must show that option 1 is the most desirable for Experiment 4. As discussed previously, it will be advantageous for the method to require as little information as possible. Second, the numerical experiment must show that the first option does indeed produce the desired characteristics in the reliability output. To accept the hypothesis, option 1 must be deemed to capture these characteristics in a more consistent manner than the other options. Finally, the first option must perform at least second best in terms of required evaluation time. In the event that the option finishes second or third best the relative differences in runtime will be evaluated. If the differences are significant, meaning an order of magnitude, then the hypothesis may need to be rejected. However, if option 1 finishes last in terms of evaluation time but the relative difference in time is small, the hypothesis can still be accepted.

3.11.2 Experiment 4 Results

After completing the assumptions for Experiment 4, the reliability growth models were run for all 288 vehicle architectures. In order to observe the trends from each option, the mean reliability values for each vehicle have been plotted below. The first option output can be seen in Figure 28.

The output for option 1 shows a large reliability growth trend with a narrowing range as the equivalent flight number increases. At the first flight the architectures have a very large range of reliability values, between 0.65 and 0.98. An interesting trend can be seen near the top most curves at the lower flight numbers.

In this portion of the output plot a few groups of reliability growth curves can be seen, which can be traced back to the options from the matrix of alternatives. These bands represent discrete jumps in reliability when selecting number of engines and number of boosters. The individual growth curves within each band represent the selections for engine out and subsystem redundancy.

The groupings seen in the option 1 output is a promising trend when considering the prediction accuracy criterion for Experiment 4. In this example case, option 1 has produced very noticeable differences between the various vehicle architecture options, which will allow the analyst to assess the effects of each option. Ultimately, the proposed approach is aiming to support this sort of decision making during early design.

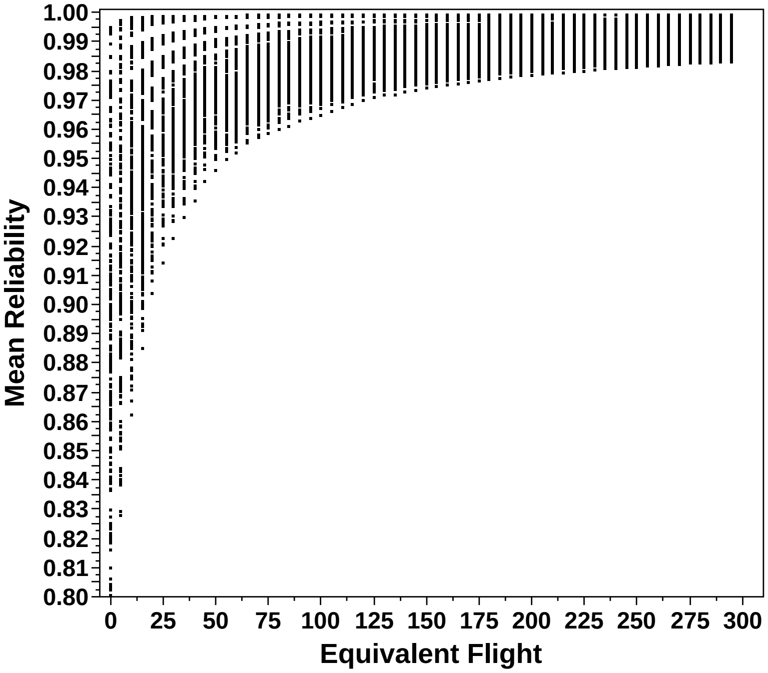


Figure 28: Reliability growth output for option 1

The output for option 2 can be seen in Figure 29 below, which plots the output on the same scaled axes as option 1. In this case it is immediately clear that the output is drastically different. The option 2 output shows a similar range in the initial reliability of the vehicle, however it immediately jumps to a higher reliability and remains within the same range for the rest of the flight history. In this case it is much more difficult to differentiate between the various architecture options as they all lie nearly on top of one another.

The very immediate reliability jump is also cause for concern as this trend is typically not seen in real systems. The large jump in reliability for option 2 may be caused by the incrementing scheme that was introduced into the growth models. Due to the fact that option 2 increments based upon assumed failure, a double counting effect may exist within the model. For each of the individual components, reliability growth projections are created at the beginning of the model run. Next, the random draws are used to determine the rate at which each of these reliability growth projections are incremented in time. The growth projections themselves, however, already include these assumed failures. Therefore, it is suspected that at every increment determined by the random draw a much larger increase in reliability will be observed.

The output of option 2 shows that the increment based upon assumed failure approach is not appropriate for application in the CONTRAST method. The very large jump in reliability along with the constant range in reliability across all flights does not bode well for prediction accuracy. If this option was implemented it would be very difficult for the analyst to find appreciable differences between the architecture options.

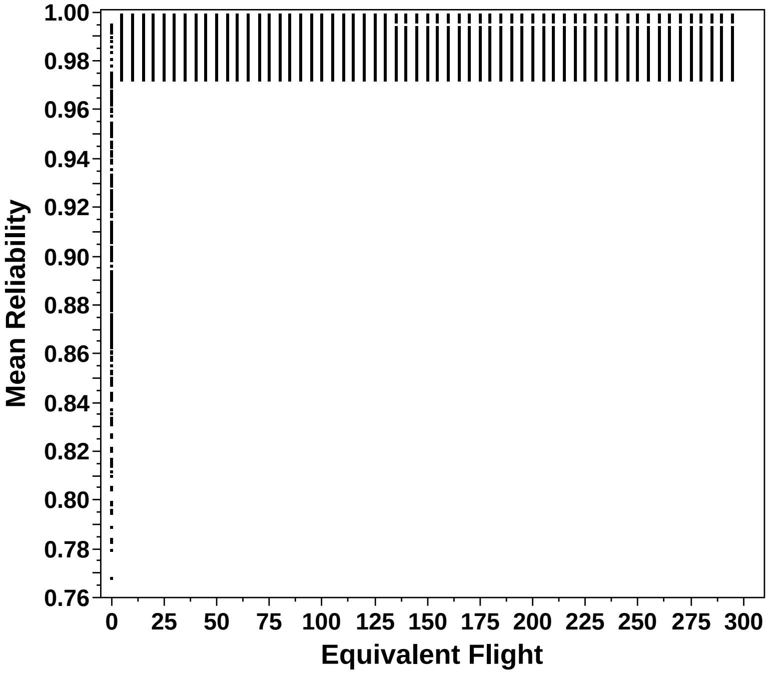


Figure 29: Reliability growth output for option 2

The final option for Experiment 4 is to anchor each of the components based upon an assumed first flight. This assumed first flight was meant to represent the amount of development time that the subsystem would go through prior to the first operational launch of the vehicle. Note that this assumption essentially means that the first flight shown in the overall system output is representative of the expected reliability at the first operational flight. This is illustrated in Figure 30 below, which shows a much more narrow range than options 1 and 2 for the reliability at equivalent flight 1.

The results of option 3 show a pronounced reliability growth trend with a narrowing range as the equivalent flight number increases. This is similar to the trend seen in option 1 but with a smaller range in the reliability values. The smaller range is most likely due to the anchoring assumptions, which effectively eliminate the earliest sections of the individual subsystem reliability growth curves. The early sections of these curves will have the largest range in reliability with a decreasing range as the equivalent flights increase. The elimination of the widest ranges leads to a smaller range at the system level.

In comparison to option 2, option 3 performs much better. This option shows a noticeable reliability growth trend for the vehicle, which is much more realistic than the immediate jump from option 2. However, option 3 does eliminate the section of the reliability growth curve that relates to the developmental period leading up to the first operational flight. As noted in the development of the research objective, the reliability growth through this developmental period is of interest to the reliability analyst. This section of the reliability growth projection is crucial to the estimation of the number of required equivalent flights to reach a specific reliability target. Ideally, the analyst can use such an estimate to determine at what point the first operational flight will be appropriate. Since option 3 sets a specific point for the first operational flight, this approach is not possible.

Another issue with option 3 is in regard to the data availability criterion for Experiment 4. This criterion was included in order to account for the extra information that is needed for each of the identified options. For option 1 and option 2, no additional information is needed besides the reliability growth assumptions for each of the subsystems. Option 3 however, requires a timeline for the development of each of the subsystems. This timeline is used to determine the anchor point for the first equivalent flight of each subsystem. Although the subsystems are typically developed at different times and rates, this assumption requires more information regarding the development schedule. In order to produce the first flight assumptions for each subsystem the vehicle development schedule must be assumed a priori. For most new vehicles, especially novel concepts, there will not be previous vehicle programs on which the assumptions can be based. This will ultimately lead to inaccuracies in the reliability growth projection for the new vehicle. It will also commit the analyst to one number for the first operational flight, which cannot be adjusted later on during the analysis. This will reduce the flexibility and overall usefulness of the results.

Overall, option 3 performs better than option 2 in terms of the relative accuracy criterion. It is not expected to perform better than option 1 for this criterion. In terms of data availability, this option is the only one that requires additional data. Therefore, option 3 is the only approach that will be negatively affected by the lack of appropriate data.

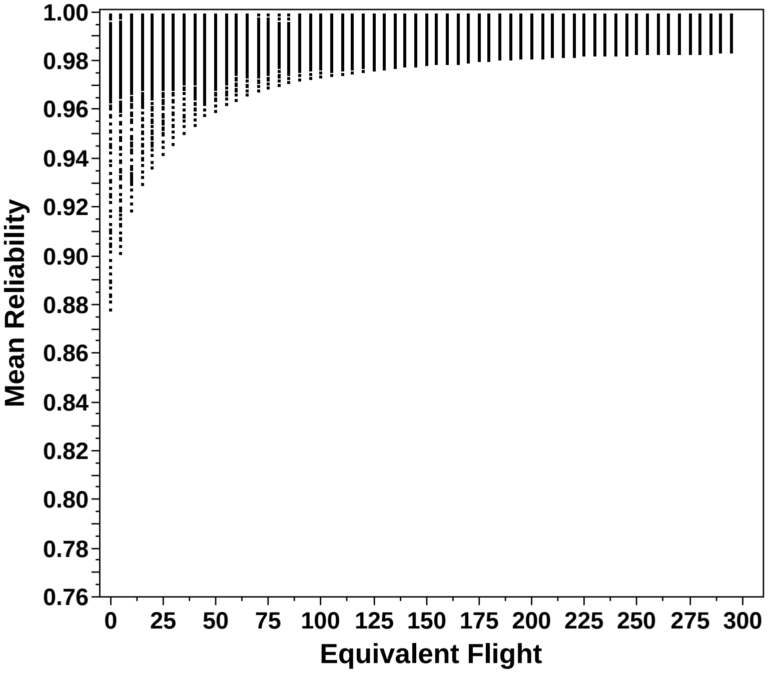


Figure 30: Reliability growth output for option 3

The output plots discussed above were used to evaluate the performance of each of the options relative to the accuracy and data availability criteria. The final criterion to be assessed is the required runtime of each option, which is an important criterion because of its implications when evaluation very large architecture spaces. A minimal runtime is desired for the method to enable the evaluation of large trade spaces.

Table 19 gives the required runtime of each of the options. This table shows that the second option performs best in terms of required runtime, with option 1 ending up in second. The third option required about 5 minutes longer to evaluate the 288 architectures than the second option. The difference between option 2 and option 1 is much smaller at 3 minutes. In terms of time required per architecture, this difference is only about 0.5 additional seconds per case. Although option 2 performs better than option 1 in terms of runtime, the discussion of the relative accuracy above can be used to eliminate option 2 from realistic consideration.

Table 19: Required runtime of each option in Experiment 4

Option	**Runtime (s)**
Anchor at flight 0	2531
Increment using assumed failure	2355
Anchor based on assumed schedule	2681

The reliability growth results for Experiment 4 evaluated three options against three criteria. In looking at the output for each of the options, the first option can be considered most desirable for the first two criteria, relative accuracy and data availability. Option 1 does not perform best in terms of required runtime, but it does come in close second to option 2. Option 2 cannot realistically be applied in the method due to its shortcomings in terms of relative accuracy. Therefore, the results of Experiment 4 can be used to accept hypothesis 6, which identified option 1 as the most desired for application in the CONTRAST method.

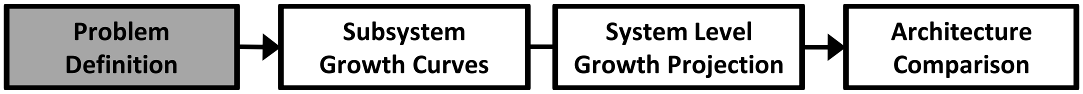
CHAPTER IV

CONTRAST METHOD

4.1 Method Description

Chapter 3 presented the development of the specific steps within the CONTRAST method. Beginning with a generic process outline the method was constructed, resulting in four primary blocks. First, the Problem Definition block represents the identification of alternatives and metrics of interest. The next block, Subsystem Growth Curves corresponds to the generation of growth projections for each of the subsystems within the defined architecture space. After producing these projections the third block, System Level Growth Projection, utilizes an automatically generated fault tree for the vehicle being analyzed. Finally, the Architecture Comparison block represents the evaluation of the method output, which ultimately leads to the selection of a baseline vehicle using reliability and safety as a figure of merit. Within the four primary blocks, five specific steps for carrying out the CONTRAST method can be identified. These steps will be discussed in more detail in the following sections.

4.1.1 Step 1: Vehicle Definition



The first step of the CONTRAST method is to identify the vehicle concepts to be analyzed. This will be achieved via the use of a morphological matrix containing all possible alternatives in the architecture space. The analyst must first break the launch vehicle into specific physical attributes such as number of stages or number of engines, and identify the potential options for each attribute. The list of physical attributes can typically be easily generated for a launch vehicle.

After determining the concepts to be evaluated, the metric of interest for the analysis must be selected. The metric of interest, such as LOM, LOV, or LOC will depend upon the types of vehicles within the defined architecture space. If manned vehicles are being considered, a safety metric in the form of LOC may be the most desirable. On the other hand, if un-manned vehicles are being considered, a reliability metric such as LOM may be more appropriate. It is important to clarify the metric of interest prior to setting the reliability growth assumptions in order to ensure consistency between the subsystem models. Only failure modes that contribute directly to the metric of interest should be included in the model assumptions.

Next, the analyst must prepare the metadata to be used in step 3 of the CONTRAST method. This data can be prepared in two different ways depending upon the heritage of the subsystem under consideration. The first approach involves subsystems that have extensive flight history or are derivatives of subsystems with extensive history.

In this case it is very likely that design, test, and operational data will be available. Ideally, existing FMEAs can be used to estimate the number of failure modes inherent to each subsystem. These FMEAs can also be used to generate the probability of occurrence assumptions for the growth models if criticality was included in the analysis.

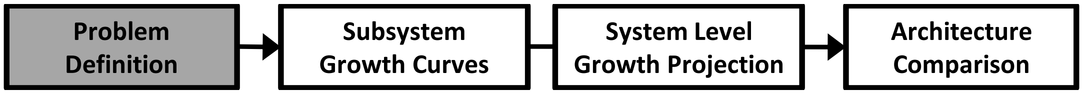
If FMEAs do not exist for the subsystem a simple PCM approach can be used based upon any other design specifications or analysis. This parts count approach will generate a generic list of failure modes inherent to the subsystem. In the absence of FMEA data, the probability of occurrence assumptions can be generated based upon previous test or operations data. If the subsystem benefits from extensive flight history, it is likely that estimates for its overall reliability have been calculated. In this case, it is suggested that the subsystem reliability be set to the value derived from the data. Then the probability of occurrence distribution can be derived by assuming that the sum of the probabilities of occurrence of all the subsystem's failure modes must equal one minus the subsystem's reliability. This approach was demonstrated in the example problem and is outlined in Appendix C.

The second approach for generating the reliability growth assumptions comes into play when dealing with subsystems that are new or novel concepts. In this case, there are no similar subsystems to compare to and no historical data to utilize. For a new subsystem, the PCM approach discussed above must be employed using a generic description of the subsystem. This generic description simply requires a layout of the subsystem that identifies the major components within the subsystem.

The probability of occurrence assumptions for the failure modes of a new system must be developed based upon the literature. Ideally, a similar historical system could be used. However, exotic concepts will not be able to rely upon such data. In this case it is suggested that the generic reliability distributions discussed in Experiment 1 be used. These distributions can be used for the assumed probability of failure of the subsystem as a whole, which allows for the derivation of the probabilities of occurrence for the failure modes. The example problem presented in the following section will illustrate the merit of this approach.

Once the metadata is populated the matrix of alternatives is complete. The vehicle definition can now be carried out by selecting one option for each row of the MOA. After a full vehicle concept has been identified, the metadata from each selected option will be collected. From this the second step, fault tree generation, can be carried out.

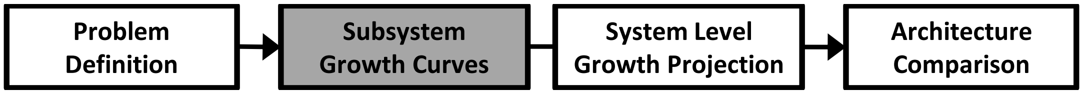
4.1.2 Step 2: Fault Tree Generation



Based upon the selected options in the MOA, the fault trees will be automatically generated using a basic set of rules. The fault trees will be set up in a relatively simple manner, making automatic generation fairly straight forward. Basic events will be defined for each option that is selected from the MOA. These basic events represent the failure of the specific subsystem. For example, if the analyst selected engine option 1 the fault tree would include a basic event representing the failure of that engine. Additional parameters will determine the number of basic events produced for each option. For example, if four engines were selected along with engine option 1, four basic events for engine failure will be produced. Another example is the use of redundancy. If a fully redundant power subsystem is included, the fault tree will include two basic events for power subsystem failure connected to an AND gate. In the case of redundancy, a common cause failure event is also included in the FTA setup. If common cause failures are not being considered, the CCF fraction for each component can be set to zero.

In addition to creation of basic events, AND gates, OR gates, and k-out-of-n voting gates will be included in the tree to link the top level event to the basic events. The top level event of interest for launch vehicles will be the probability of loss of mission (LOM), loss of vehicle (LOV), or loss of crew (LOC). With the creation of each basic event, the metadata from the corresponding option in the MOA will be saved. This will facilitate the creation of subsystem level reliability growth curves in the next step.

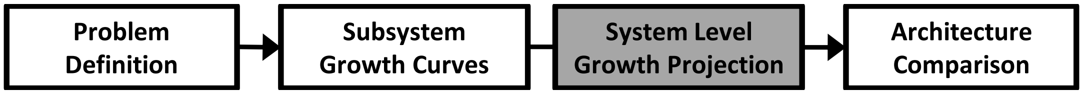
4.1.3 Step 3: Reliability Growth Curve Generation



After the fault tree has been generated, the reliability growth curves are needed at the subsystem level. These curves will be generated for each subsystem and applied to the corresponding basic event included in the fault tree. In order to create the curves, the metadata must first be accessed. This data will set up the basic assumptions for the reliability growth model. The Python objects shown in Appendix A represent the matrix row and component objects that store the reliability growth assumptions.

For each component object the "HallReliabilityGrowth" method is called. The method requires three primary inputs, which represent the number of steps in time to evaluate the model, the number of repetitions at each step, and the total length of the flight history. This method then runs a Monte Carlo simulation using the Hall model, which produces the reliability growth curve along with its confidence bounds. At the conclusion of the run of the "HallReliabilityGrowth" method, the reliability growth data is stored within the "relArray" attribute for the given component object. The implementation of the Hall growth model in Python is given in Appendix B.

4.1.4 Step 4: Monte Carlo Simulation



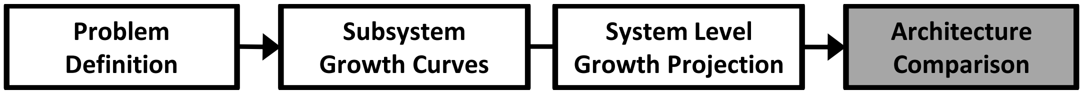
The next step in the CONTRAST method is to utilize Monte Carlo simulation to combine the lower level reliability growth curves into a system level estimate. Due to the fact that the automatically generated fault tree will be relatively simple, the exact expression for evaluation of the fault tree will be derived. This expression will include the probabilities of failure for each of the subsystems (basic events) in the tree. As was discussed in Section 3.11 a single strategy for anchoring the subsystem growth curves has been identified. This approach anchors all of the subsystem growth curves at equivalent flight 0 and increments them all at the same rate.

At every step in time the probability distributions for each subsystem are queried using the “RelUpdate” method. This method is passed the repetition number along with the step number and returns a reliability value for the subsystem at the given step in time from the “relArray” attribute. After the “RelUpdate” method is run for each subsystem within the fault tree expression, the equation can be evaluated to give the system level reliability estimate. Running a MC at one point in time will thus generate a probability of failure distribution for the system as a whole. The MC process is repeated at specific increments through time, ultimately producing a reliability growth curve with confidence bounds for the entire system.

It is important to note that the number of repetitions performed at each step in time is an important setting to be determined by the analyst. The number of repetitions is the number of random draws to be taken from the subsystem reliability distributions in order to produce the system level distribution at the given step in time. Therefore, this parameter will have a large effect on the overall runtime of the method. It will also affect the granularity of the resulting system level output. Using a very small number of repetitions will not accurately resolve the resulting system distribution, while too many repetitions will require a very long runtime for each vehicle. Appendix D presents a short study, which gives recommendations as to the number of repetitions that should be used to balance these two effects. The results of this study suggest that the number of repetitions should be limited to less than 2000 to keep the runtime at a reasonable level.

With the conclusion of the MC runs the reliability growth projections for all of the vehicles in the matrix of alternatives will be available to the analyst. The final step in the CONTRAST approach is then to evaluate the vehicle concepts.

4.1.5 Step 5: Concept Evaluation and Selection



Finally, after the system level reliability growth curve has been generated it can be stored for comparison to other vehicle architecture concepts. This step involves the comparison of the various attributes of each growth curve to one another in order to identify a desired concept. As was discussed in Section 3.1, attributes for comparison include initial reliability, expected reliability at first operational flight, mature reliability, number of flights to minimum required reliability, and number of flights to maturity.

Although concept evaluation and selection is a major step in the CONTRAST method, a specific method for carrying out concept selection is beyond the scope of this research. The primary research objective is to produce reliability estimates that will aid the decision maker regardless of the specific decision making technique being employed. Therefore, this research will only suggest trends and various attributes that the decision maker can expect to see and exploit from the CONTRAST method. Chapter 5 will present a full application of the CONTRAST method to an example launch vehicle problem. In this chapter the comparison of different concepts will be illustrated and the key advantages of using the CONTRAST method will be explained.

4.2 Example Problem

After developing the CONTRAST method, an example problem was selected in order to demonstrate the reliability growth approach on an actual vehicle. This example problem will serve as a final validation effort to show that the CONTRAST method can accurately predict the reliability of a previous launch vehicle. The Space Transportation System, otherwise known as the Space Shuttle, was chosen for the example problem. The shuttle was chosen due to its extensive flight history, as well as availability of vast amounts of documentation in regards to design, testing, and operations.

The Space Transportation System (STS) is made up of four primary elements including the orbiter, external tank, and two re-usable solid rocket boosters. The orbiter is the element used to carry astronauts as well as any payload to and from orbit. Each orbiter was designed to be fully reusable for up to 100 missions [115]. Over the course of the Space Shuttle program, five orbiters were flown; Columbia, Challenger, Discovery, Atlantis, and Endeavor [27]. Although the orbiter supported all in-space and re-entry activities, it also served an important purpose at launch, housing three Space Shuttle Main Engines (SSMEs).

The SSME is a re-usable staged-combustion cycle liquid rocket engine that utilizes liquid hydrogen and liquid oxygen as fuel and oxidizer, respectively [133]. On launch the three SSMEs on the orbiter were burned for approximately 520 seconds and were the sole source of thrust after solid rocket booster (SRB) jettison [115]. The fuel and oxidizer for the SSMEs was stored in the external tank element of the STS.

The external tank was the only non-reusable primary element of STS. During each launch the tank was jettisoned and burned up upon re-entering the earth's atmosphere. The external tank was used to store liquid hydrogen and oxygen, which was cross-fed into the orbiter to be used by the SSMEs.

A pair of re-usable solid rocket boosters attach to the external tank, which were the primary source of thrust for the first segment of the launch trajectory [115]. During launch the SRBs are jettisoned after approximately two minutes of flight, ultimately making parachute assisted ocean landings. After the SRBs made water landings they were recovered and refurbished for future launches. Each SRB was considered reusable for up to 20 launches [115].

Since the example problem will only utilize a single vehicle architecture the fault tree for the STS can be generated prior to the execution of the growth models. This fault tree will include basic events for each of the identified subsystems within the vehicle architecture. The development of the fault tree and its equations is presented in the following section.

4.2.1 STS Fault Tree

To carry out the example problem, the STS was first broken into its four primary elements. These elements will serve as the primary "subsystems" that make up the STS launch system. The SRBs and external tank will not be broken into further subsystems for the example problem. However, the fourth STS element, the orbiter, will be further decomposed into vehicle subsystems. The reason for this decomposition is the inclusion of multiple mission critical subsystems into the orbiter element. These subsystems include the SSMEs as well as the Shuttle avionics, for which reliability growth curves will be generated. After identifying these elements the fault tree for the STS can be manually produced.

The STS fault tree used for this example problem can be seen in Figure 31 below. The fault tree begins with the top level LOC event. Loss of crew was chosen over loss of vehicle due to the availability of STS data. In reference [78], Hamlin presents the reliability growth of the shuttle in terms of probability of LOC at each mission. This data was generated using the full STS probabilistic risk assessment, which is accepted as the state-of-the-art for estimating mission risk [121, 129].

From the top level event, four failures representing the four primary elements of STS are connected via an OR gate. Each SRB and the external tank are represented using circles, which are basic events. This indicates that the three elements will not be decomposed any further.

The fourth event in the second level of the tree refers to an orbiter failure that will cause an LOC event. In this case the orbiter is decomposed into three sub events, avionics failure, structural failure, and SSME failure. These events are connected via an OR gate. The avionics failure is a basic event and will be represented using a reliability growth curve. The structural failure event is also a basic event, but will be represented using a simple probability based upon the literature. The vehicle structures are expected to mature at a very rapid rate compared to the other subsystems. Therefore, the structures will be assumed to be at a "mature" reliability point prior to the first test launch of the vehicle.

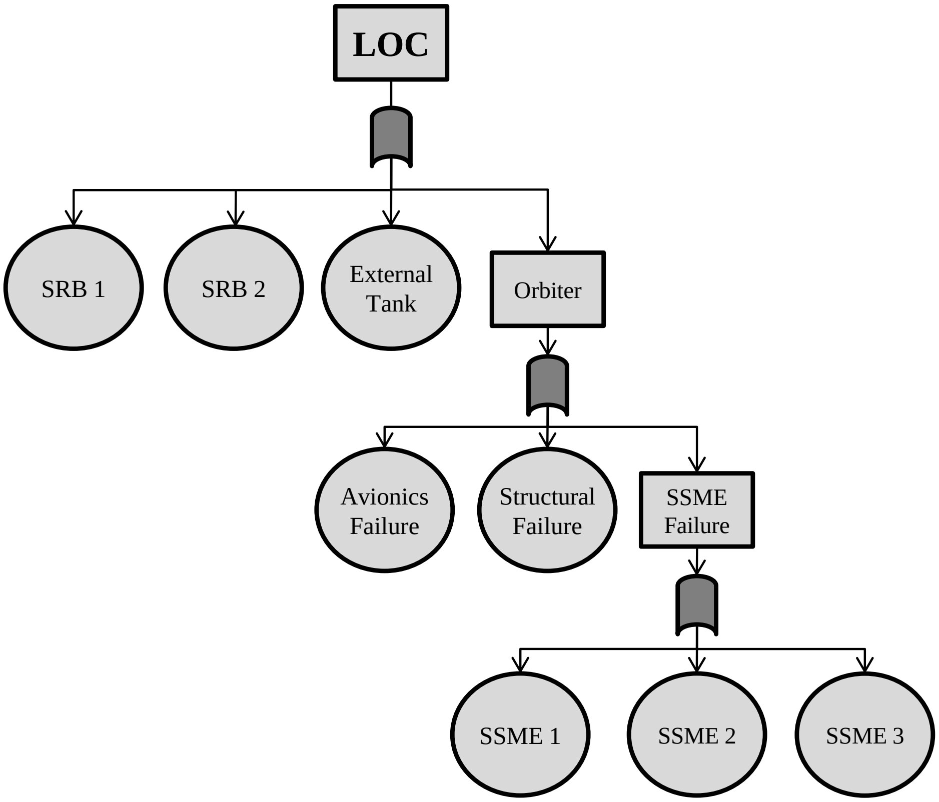


Figure 31: High Level STS fault tree

Due to the fact that the orbiter houses three SSMEs, the third orbiter failure event is decomposed into three individual SSME failure events. It is important to note at this point that the individual SSME failure events are linked via an OR gate and not a k-out-of-n voting gate. A k-out-of-n voting gate would represent the case where two out of the three SSMEs would have to fail in order to cause an orbiter failure. This translates to the existence of an engine out capability, which the STS did have [74].

However, this engine out capability is only applicable to benign engine failures. Due to the LOC top level event, the SSME failures being considered are of the non-benign type, which can cause damage or destruction of the other SSMEs, elements, or the entire vehicle. For this reason a standard OR gate is used and the reliability growth curves for the SSMEs are generated under the assumption that all failure modes are catastrophic in nature.

After generating the STS fault tree, a mathematical expression for the probability of LOC can be derived. This expression will be used in addition to the subsystem level reliability growth curves to generate the expected growth curve for the full STS vehicle. From the top level event, the probability of LOC can be written:

$$P_{LOC} = 1 - (1 - P_{SRB1}) * (1 - P_{SRB2}) * (1 - P_{ExternalTank}) * (1 - P_{Orbiter}) \qquad (27)$$

where $P_{Orbiter}$, and P_{SSMEs} can be written:

$$P_{Orbiter} = 1 - (1 - P_{Avionics}) * (1 - P_{Structures}) * (1 - P_{SSMEs}) \qquad (28)$$

$$P_{SSMEs} = 1 - (1 - P_{SSME1}) * (1 - P_{SSME2}) * (1 - P_{SSME3}) \qquad (29)$$

In Equation 28, the value for $P_{Structures}$ will be defaulted. This parameter represents the probability of failure of the orbiter structures. As was discussed previously, this probability was estimated based upon data published by Hamlin [78]. The value used for the probability of orbiter structural failure for the example problem is 1 in 340 flights.

The remaining probabilities in Equations 27, 28, and 29 will be generated based upon projected reliability growth curves. These probabilities include P_{SRB1}, P_{SRB2}, $P_{Avionics}$, and P_{SSMEs}. The following section will discuss the generation of the reliability growth curves and enumerate their underlying assumptions.

4.2.2 STS Reliability Growth Assumptions

In Section 3.4 of Experiment 1 the Hall growth method was identified as the most suitable for application in the CONTRAST method. Therefore, the Hall method was used to generate the necessary reliability growth curves for the example problem. As discussed in Section 3.3.1, the Hall model contains three primary assumptions: number of failure modes, probabilities of occurrence for each mode, and a fix effectiveness factor for each mode. In Section 3.5 the level of application of the Hall model was discussed, resulting in the identification of the subsystem level as the most appropriate. Although the assertion to research question 3 identified this level as best, both the system and subsystem level approaches will be included in the example problem. The use of the system and subsystem level approaches in the example problem will further illustrate the difference between the two. The subsequent sections will develop the reliability growth assumptions at the system and subsystem levels for the STS.

4.2.2.1 System Level Assumptions

The system level assumptions for the STS are very simple to produce for the example problem because they have already been addressed by Experiment 1 in Section 3.4.1. Within Section 3.4.1 the number of failure modes for the system level approach was defined as a range between 6 and 12 modes. This range was determined based upon the number of basic events that show up in the STS fault tree that contribute directly to an LOC. In Figure 31 there are eight such basic events, which lies within the defined range for number of failure modes.

The system level probability of occurrence distribution was also defined in Section 3.4.1 for the STS vehicle. In this section the probability of occurrence distributions given in the literature were compared. Ultimately, it was shown that multiple authors presented similar probability of occurrence distributions. One of these distributions was chosen from the literature to represent the probability of occurrence values for the STS system. This distribution is Beta(0.25,8.75), which has a mean of 0.025 and a standard deviation of 0.05.

The final assumption for the system level approach is the fix effectiveness factor. These factors were discussed in more detail in Section 3.4.1 for the system level approach. In this section it was concluded that a relatively high fix effectiveness factor was appropriate for the STS. This is primarily due to the fact that the STS is a manned vehicle, which requires very extensive post flight data analysis and anomaly investigation. The fix effectiveness factors for the system level were therefore modeled using a uniform distribution between 90% and 99%.

4.2.2.2 Subsystem Level Assumptions

After generating the assumptions for the system level approach, the subsystem level assumptions must be addressed. First, the individual subsystems of the STS must be defined. Then the number of failure modes, probability of occurrence, and fix effectiveness factors must be set for each of the subsystems.

The STS consists of four primary elements, the orbiter, solid rocket boosters, and external tank. The orbiter houses three primary subsystems that are essential to the success of an STS launch, the primary avionics, power, and Space Shuttle Main Engines. Figure 31 illustrates the breakdown of the STS subsystems. From this figure four individual subsystems can be identified for the example problem. These subsystems include the orbiter avionics and power, SSME, solid rocket booster, and external tank.

The first subsystem that will be considered is the Space Shuttle Main Engine. The SSMEs are staged combustion engines that burn liquid oxygen and liquid hydrogen. In order to produce an estimate for the number of failure modes of the SSME a simple parts count approach can be utilized. This PCM approach is based off of the notional diagram of a staged combustion engine given in Figure 32. Figure 32 shows a notional diagram of a staged combustion engine, which includes turbopumps for the fuel and oxidizer, a pre-burner, flow control valves, a combustion chamber, nozzle, and heat exchanger. In addition, the engine may include a hydraulic system for thrust vector control. Although the notional engine shows only a single pre-burner and one turbopump each for the fuel and oxidizer, multiple pre-burners and turbopumps may be used. In fact, the SSME utilizes two turbopumps for each propellant along with two pre-burners [155]. From the notional diagram an estimate was generated of between 8 and 15 failure modes for the SSME.

In order to support the above approach for generating the number of failure modes assumptions, actual data for the SSME was gathered. Due to the extensive testing of the SSME throughout the STS program, a wide variety of failure analyses, failure reports, and FMEAs are available. One source in particular was used to evaluate the number of failure modes assumption from above.

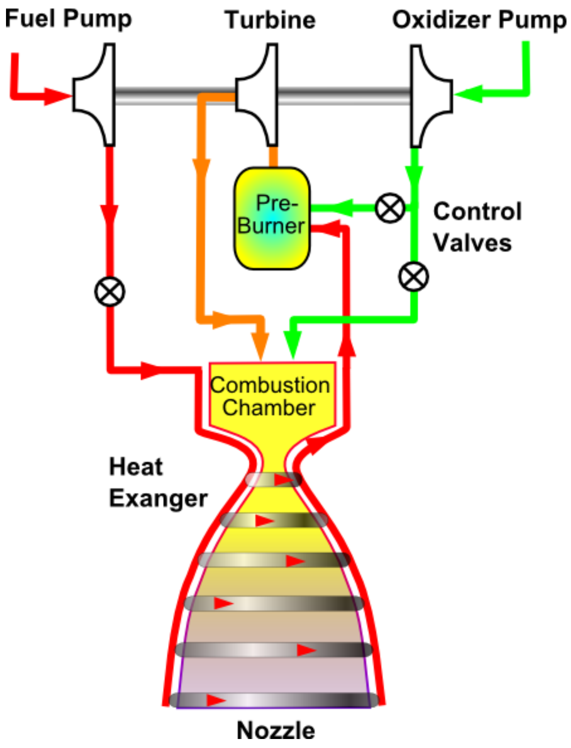


Figure 32: Notional diagram of a staged combustion liquid rocket engine

This source is a comprehensive FMEA that was developed for the SSME in the mid-1980s [70]. Within this FMEA around 190 total failure modes were identified and evaluated for the SSME. Each of the modes were categorized based upon criticality and the estimated end effect. Although 190 is a large number of modes, only 3 of these modes were determined to cause an immediate loss of the vehicle and crew [70]. An additional 7 failure modes were labeled as probable loss of vehicle. The next lower criticality level, loss of engine, includes 3 more failure modes. Conservatively speaking, the total number of modes from the detailed FMEA that could lead directly to an LOC is 10 or 13 depending upon the inclusion of the loss of engine criticality level. In comparison to the PCM approach, this number lies directly within the defined range for number of failure modes for the SSME.

The next element of the STS to consider is the external tank (ET). The ET houses the liquid oxygen and liquid hydrogen used by the SSMEs during ascent. The tank also supports the entire system during ascent, serving as the primary attach point for both boosters and the orbiter. A list of primary components of the external tank is given below. From this list a range of 7 to 15 failure modes for the external tank was derived.

- Liquid hydrogen tank: 2 domes, one cylinder
- Liquid oxygen tank: 2 domes, multiple cylinders
- Intertank SRB attach points
- Orbiter attach points
- Orbiter cross-feed lines
- Internal anti-slosh baffles
- External foam insulation

The next element to be considered for the STS is the solid rocket booster. As shown in Figure 20 the STS system utilizes two SRBs. The SRBs are four segment boosters using a mixture of primarily powdered aluminum and ammonium perchlorate as propellant and oxidizer [115]. The boosters also house instrumentation for thrust vector control, separation, and recovery. For the ascent phase of the STS mission a list of key booster components is given below. From this list a range of 7 to 15 failure modes per booster was derived for use in the example problem.

- Four propellant segments including casing
- Joints at each segment connection
- Nozzle and vector control
- Igniter
- Pad tie-down and vehicle supports
- Connector struts to ET

The final subsystem to consider for the STS example problem is the orbiter avionics. This subsystem is assumed to include the power subsystem of the orbiter as well. The orbiter avionics subsystem controls all tasks regarding guidance, navigation, and control during vehicle assent. It also includes communications between the vehicle and ground control and the timing of staging events such as SRB separation. Similar to the previous subsystems a list of key components and functions was put together for the avionics subsystem. This list can be seen below and was used to develop the range of failure modes for the avionics subsystem. The range of number of failure modes for the avionics subsystem was set between 10 and 20.

- Flight computers (hardware and software)
- Data handling/processing

- Instrumentation (flight controls, sensors, etc.)
- Power systems: Fuel cells, auxiliary power units
- Power distribution
- Cabling and wiring for flight controls, sensors, etc.

After identifying the number of failure modes assumptions for each of the STS subsystems, the probability of occurrence values for the modes must be defined. Since the system and subsystem level approaches will be compared in the example problem it is important to setup the probabilities of occurrence to be consistent between both levels. In order to ensure consistency, the probability of occurrence distributions for the subsystems were derived based upon the system level distributions. The same approach was taken in Section 3.11.1 of Experiment 4 resulting in the Beta parameters given in Table 17. In this approach, the number of modes for each subsystem determined the number of random draws from the subsystem Beta distribution. The summation of these draws was then required to approach the Beta distribution defined for the system level. This process was carried out numerically using Equation 26 until the subsystem Beta parameters were converged upon.

Based upon the maximum defined number of failure modes for each STS subsystem, this approach was used to determine the Beta parameters for the probability of occurrence distributions. Table 20 below gives the maximum number of failure modes and the resulting Beta parameters for each of the STS subsystems. Since the maximum number of modes is the same for three of the four subsystems, only two unique Beta distributions were used for the probabilities of occurrence at the subsystem level.

Table 20: Parameters for STS subsystem failure mode probabilities of occurrence

Subsystem	Max # of Modes	Beta parameters (a,b)
SSME	15	0.02, 10
External Tank	15	0.02, 10
SRB	15	0.02, 10
Avionics	20	0.015, 11

The final assumption that must be derived for the subsystem level growth models is the fix effectiveness factor. As with the probability of occurrence assumptions, the fix effectiveness factor assumptions must be set in order to ensure consistency between the system and subsystem approaches. Therefore, the same range for fix effectiveness will be used for all of the subsystems. The fix effectiveness factors for the subsystems will thus be uniformly distributed between 90% and 99%, which is the same as the system level approach.

4.2.3 STS Example Problem Results

After completion of the reliability growth model assumptions the models were setup for the STS system and underlying subsystems. For every application of the growth model 135 flights were assumed and the model was evaluated at every flight. Note that for the subsystem level approach, the reliability growth models were applied to the system level fault tree according to the results of Experiment 4. This means that all subsystem growth curves were assumed to be anchored at equivalent flight 0. The results from both the system and subsystem levels can now be compared against the PRA data. The PRA data for the STS was discussed in more detail in Experiment 1 and can be seen in Figure 21.

Figure 33 below shows the system level reliability growth results plotted against the PRA data. As can be seen in the plot, the system level approach is very close to the actual STS data. The mean predicted value tends to be greater than the mean value given by the PRA. However, the mean prediction still lies within the 95th percentile of the PRA data. The 5th and 95th percentiles of the prediction fully encompass nearly all of the mean values of the PRA data except for one point around flight 90.

At this point in the flight history, the actual data takes a slight dip in reliability. This reduction in reliability is due to a block upgrade that was performed during the shuttle program. In the development of the PRA reliability growth data for the STS, Hamlin notes that this specific reduction in reliability was due to a change in the process for applying foam insulation on the external tank [78]. A similar effect is seen during the early flights between 10 and 20, which was due to the disabling of the ejection seats in the orbiter [78].

Overall the system level results perform well in comparison to the PRA data. The range between the 5th and 95th percentiles is fairly narrow for the system level approach, but it still captures a majority of the range shown by the actual data. The only possible issue to note is that the results tend to over predict the reliability of the vehicle.

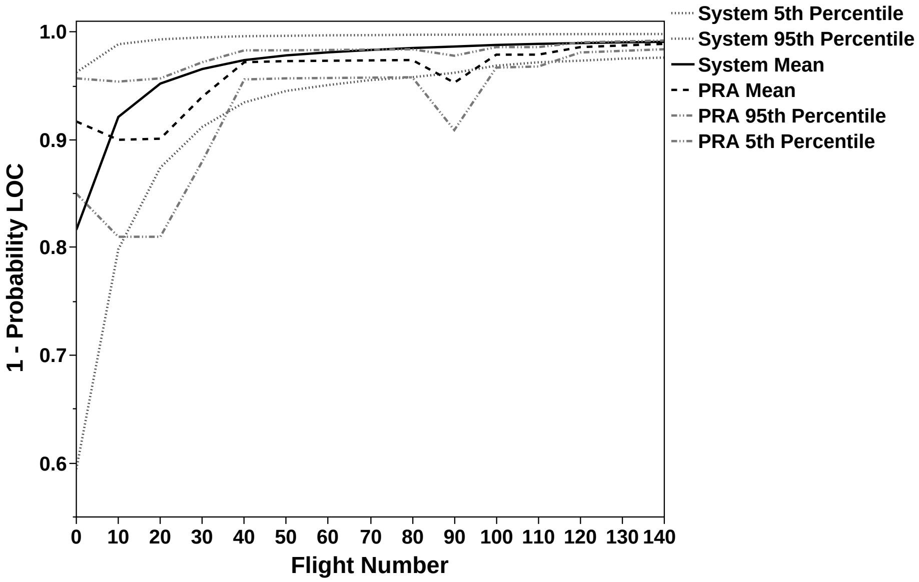


Figure 33: System level reliability projection versus STS PRA data

The subsystem reliability results can be seen in Figure 34 along with the PRA data. What can be seen almost immediately is that the subsystem level results display a much wider range between the 5th and 95th percentiles. Although this is generally undesirable, it does allow the subsystem results to almost fully capture the entire range of the PRA data. Only two points fall outside of the subsystem results range, both are on the 5th percentile line of the PRA output. From this figure it is also apparent that the growth prediction is no longer over predicting the STS reliability. The mean value of the subsystem results follows the mean value of the PRA data fairly well but tends to be on the low side of this data.

Figure 34 shows the validity of the CONTRAST method for projecting launch vehicle reliability growth. Using a fairly simple set of assumptions stemming from the literature and parts count, the reliability growth behavior of the STS vehicle has been fully captured by the CONTRAST method. The wide bounds shown by the results are not necessarily a detriment to the prediction accuracy either. The bounds ensure that any behavior that the vehicle may encounter will be encompassed within the prediction. This behavior may include discrete positive or negative jumps in reliability due to upgrades or plateaus in reliability at any point during the flight history.

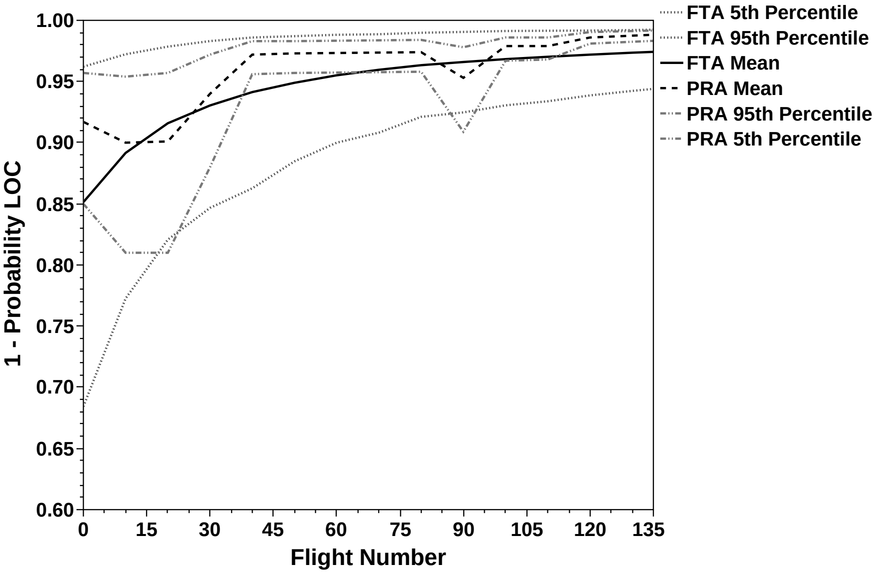


Figure 34: Subsystem level reliability projection versus STS PRA data

After comparing the system and subsystem results to the PRA data a few key observations can be made. First, the ranges produced by each of the approaches are very different, with the subsystem results showing a much larger range. The narrow range displayed by the system level results seems to be more desirable. However, this narrow range does not allow it to fully capture the behavior of the actual vehicle. As noted above there are multiple points that fall outside of the lower bound of the system level prediction, which is undesirable. Although the subsystem level results show a much wider range in the reliability predictions, this range allows for the full capture of the PRA data.

The second observation is in regard to the accuracy of the system level output. The mean value of the system level prediction falls very close to the mean value of the PRA data. This result is especially important when considering the probability of occurrence assumptions for the growth model. As discussed previously, the probability of occurrence values can be difficult to estimate when no historical data is available. For the example problem it was assumed that no data was available, thus the probability of occurrence values were derived from literature. A very simple distribution was setup using the reliability growth literature, which seems to have represented the actual system very well. This is a very promising result because it shows the validity of these assumptions.

Although the system level results are very promising, the discussion of research question 3 shows the need for the subsystem level approach. Recalling this discussion, the system level approach is unable to capture changes in reliability due to architecture options such as redundancy or engine-out capability. Since the example problem was looking at catastrophic failures only, inaccuracies due to redundancy or engine-out were not seen. Therefore, the results of the example problem do not change the conclusion that the subsystem level approach is more appropriate.

In addition to testing the system level approach again during the example problem, the various approaches for incrementing the subsystem level growth curves were implemented. The results of this additional test can be used to further support the conclusions drawn by Experiment 4. Figure 35 below gives the results for option 2 of Experiment 4, while Figure 36 shows the results from option 3. Recall that option 2 refers to the incrementing based upon assumed failure, while option 3 refers to anchoring of the subsystem growth curves based upon assumed schedule. For the STS example problem option 3 utilized the timeline discussed in Section 3.11.1 for the assumed schedule.

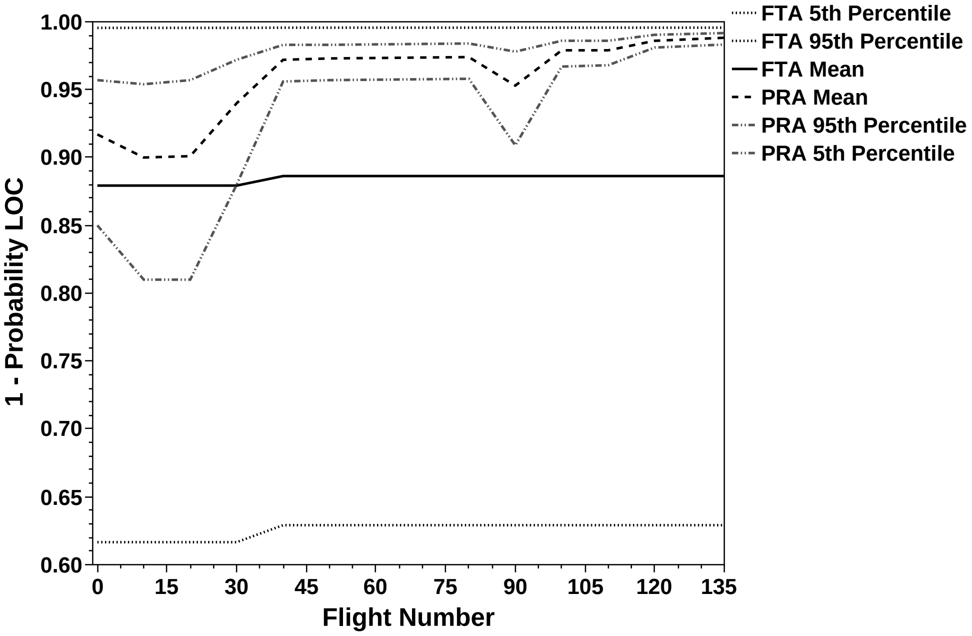


Figure 35: Subsystem level reliability projection using incrementing option 2

The reliability growth results for the example problem using option 2 can be seen in Figure 35 above. In the figure the growth projection is plotted against the STS PRA data. This figure very quickly illustrates the issues with using the alternative incrementing scheme. The range between the 5th and 95th percentiles of the predicted data is very large for the entire length of the flight history. Although the predicted data fully encompasses the actual PRA data, the prediction shows that the vehicle will lie somewhere between 0.6 and 0.99 for the entire flight history. The mean predicted value also lies well below the actual data and does not capture much of the reliability growth trend. This result is similar to what was illustrated in Experiment 4, where the resulting range of the predictions was very large. As discussed during that experiment, the increase in range is most likely due to a double counting effect. Figure 35 ultimately shows that the first option from Experiment 4 will perform much better in terms of prediction accuracy than option 2.

The example problem results using option 3 from Experiment 4 can be seen in Figure 36. In this figure the reliability growth projection is plotted against the actual PRA data for the STS. As can be seen in the figure, option 3 performs better than option 2 from above. The reliability prediction has a much more narrow range in this case. However, option 3 tends to over-predict the vehicle reliability. The mean value of the prediction lies along the 95th percentile line of the PRA data for the entire flight history. This result would suggest that the assumed first flight approach gives an overly optimistic estimate of reliability.

The increase in reliability seen from this option is due to the fact that the subsystem growth curves are anchored at non-zero equivalent flights. What this means is that the initial reliability of each of the subsystems is higher in comparison to option 1. Since the subsystem reliabilities start at a higher value, it is no surprise to see that the entire flight history shows a higher reliability. Similar to the results from option 2, the option 3 results in Figure 36 further supports the conclusions from Experiment 4. This figure shows that option 3 does not perform as well as option 1 in terms of prediction accuracy.

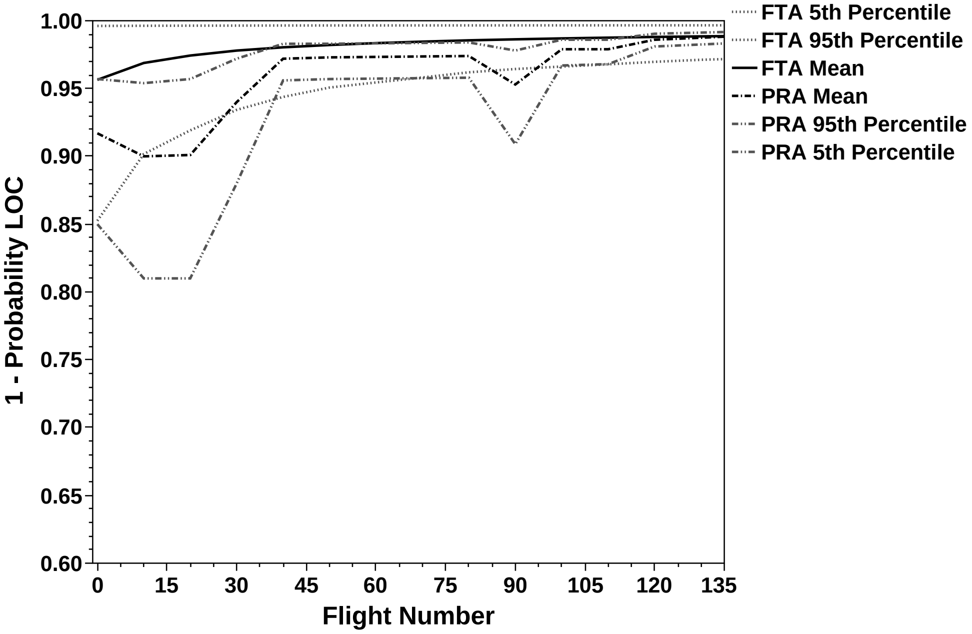


Figure 36: Subsystem level reliability projection using assumed first flight

As illustrated by Figure 34, the output from the first incrementing option fully encompasses the given PRA data. This shows that the output of the CONTRAST method is able to capture any architecture effects on reliability throughout the flight history using the specified incrementing option. However, it is interesting to note the difference between the given data and the projection from the CONTRAST method. The difference between the given data and the reliability projection is very clear; the PRA data includes discrete jumps in reliability, while the projection is continuous. Although the CONTRAST output in Figure 34 is continuous, the data still captures the discrete nature of the given data.

First, consider the given PRA data in more detail. Along with the data in reference [78], Hamlin gives a description of the primary changes that were made to the STS vehicle throughout its operational life. Each of the discrete jumps seen in the given PRA data can therefore be traced to an incremental improvement or block upgrade of one of the subsystems or components within the STS system. For example, at the beginning of the timeline a negative effect is seen in the growth history. This change represents an increase in the probability of loss of crew, which can be traced to the de-activation of the ejection seats in the orbiter [78]. Another example occurs later in the flight history around flight 90, which represents the point at which an upgraded high pressure oxidizer pump was introduced into the SSME [78].

Although these changes are very detailed and cannot be predicted during conceptual design, the CONTRAST method can still capture such results. The projection shown in Figure 34 is only continuous because it is an aggregate of 1000 total runs of the growth model. The Hall model itself is discrete, which means that each individual run of the model will show discrete jumps and plateaus in reliability.

The effects of incremental improvements or component block upgrades can therefore be traced to the discrete jumps seen in the individual growth model results. Within the results of the model the probability of mode occurrence will determine how often a discrete jump occurs and the fix effectiveness factor will define the magnitude of the change in reliability. Figure 37 shows three individual growth model runs plotted against the mean and percentile data from the STS PRA.

These cases were selected to illustrate some similarities between the PRA data and the growth model output. They also show discrete improvements that happened randomly during the model run, which line up with different discrete jumps in the PRA data. Note that each case represents a potential reliability growth track that could be taken by the vehicle throughout its flight history.

The first case, colored in red, shows a discrete improvement that aligns with the PRA data early on in the flight history. This trend, labeled A, was caused primarily due to adjustments made to the SRB in the return to flight after the loss of Challenger [78]. In this case during the growth model run, a failure mode occurred around flight 20 and due to a relatively high fix effectiveness factor the probability of loss of crew was improved from 0.90 to around 0.96. After this improvement the PRA data plateaus between flight 40 and 75.

This plateau was captured by the second case from the growth model, which is colored yellow. In the second case multiple failure modes occurred early in the flight history, causing a rapid improvement in the probability of loss of crew. After reaching flight 20 however, the probability of loss of crew plateaued at nearly the same level as the PRA data labeled B. This plateau ultimately captures a period of time where no major improvements or design changes were made to the vehicle.

The final case from the growth model, shown in green, captures the other major discrete change in the PRA data. The original downward trend at this point in the flight history was caused by the introduction of upgraded oxidizer pumps in the SSMEs [78]. The upward trend labeled C was caused by the introduction of a new foam application process for the external tank, which was supposed to reduce the risk of debris strikes to the orbiter [78]. As seen in the figure a failure mode occurred in the growth model at the same point in time as the upgrade in the PRA data. In this case, the magnitude of the upgrade is closely matched as the growth curve plateaus around 0.98.

Although the three cases shown in Figure 37 were specifically selected because their random discrete trends matched the PRA data, their output helps illustrate how the CONTRAST method can capture architecture effects on reliability. During conceptual design it is nearly impossible to anticipate that a turbopump upgrade for the liquid engines will be performed at equivalent flight 90. However, the effects of said incremental upgrade can be taken into account in the CONTRAST method. For example, an upgrade could be forced to occur at a specified point in the flight history, which would produce a discrete change in the results. The magnitude of such changes could be used to help choose between potential upgrade options. It could also be used to identify the order in which the upgrades should occur in order to maintain the highest reliability throughout the flight history. These two approaches will be demonstrated in Chapter 5, which includes analysis of potential block upgrades for the SLS vehicle.

The comparison of the individual growth model cases to the PRA data also supports the use of a simple parts count approach for developing the number of failure modes assumption. The parts count approach identifies key components within each subsystem and assigns them as "failure modes" within the growth model. Therefore, the modes within the growth model represent components that may be upgraded later on. Thus any discrete changes due to the modes in the model will capture many of the potential upgrades that will be encountered during the actual program. The probability of occurrence therefore represents the probability that an incremental upgrade or design change will be made to that specific part. The fix effectiveness factor then represents the magnitude of the improvement, or detriment of the upgrade or design change.

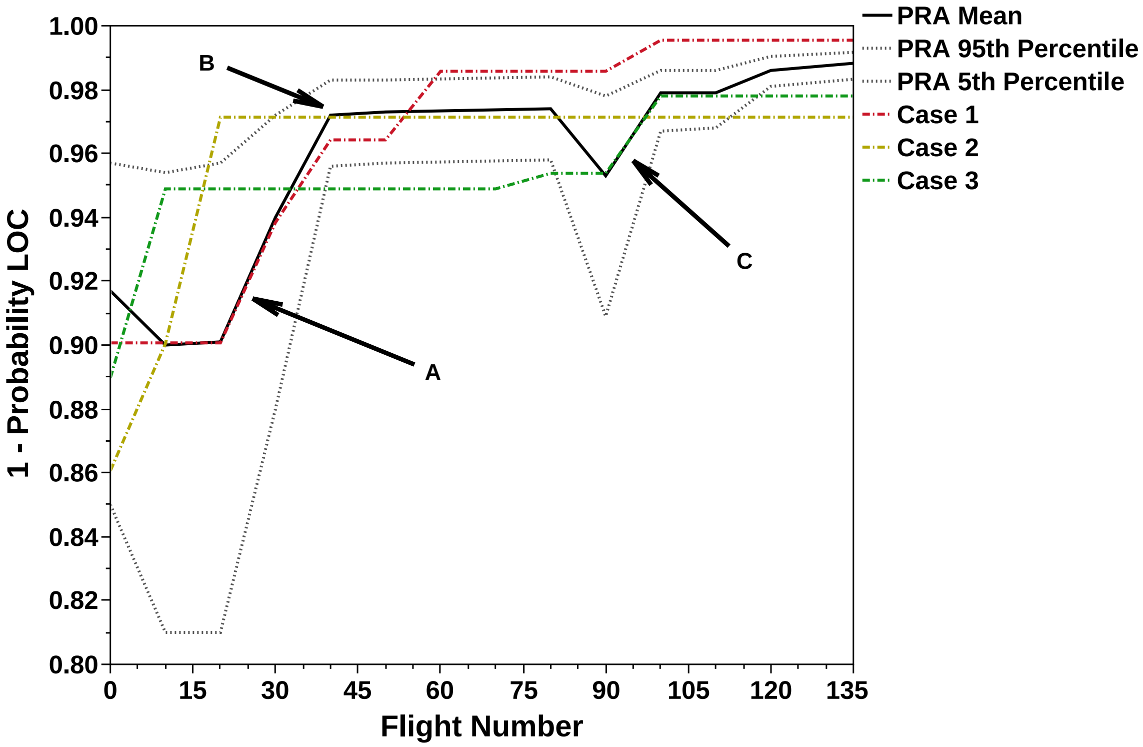


Figure 37: Subsystem level reliability projection using assumed first flight

4.2.4 Example Problem Conclusions

The STS example problem was setup as a validation exercise for the CONTRAST method. The primary goal of this example was to demonstrate the ability of the method to accurately predict the reliability of an actual launch vehicle. The example problem was also used to further support the conclusions drawn from research question 3 and Experiment 4. To conclude the example problem, a summary of the findings is necessary.

First, the example problem results illustrated the accuracy of the CONTRAST method. Figure 34 above shows the reliability growth prediction of the method plotted against the PRA data for the STS. This figure illustrates the ability of the CONTRAST method to predict vehicle reliability growth. The prediction in this figure fits the PRA data very well.

The fit of the prediction data also validates the reliability growth assumption approach used in the CONTRAST method. To setup each of the subsystem reliability growth models a parts count type approach was utilized, which simulates the complete lack of available historical data. The accuracy of the resulting reliability growth projections shows promise for the parts count approach. In addition, Figure 37 illustrates the connection between the parts count approach and the actual data. Using PCM to define the modes within the growth model enables the model to capture potential incremental changes or block upgrades that may occur throughout the vehicle's life-cycle. These results ultimately increase confidence in utilizing the parts count approach when no historical data is available for a subsystem within the identified vehicle architecture space.

Additional results for the example problem were produced in order to support the assertion to research question 3 and the conclusions of Experiment 4. For assertion 3, a system level approach was included in the example problem, which generated a growth projection for the STS at the system level. Although the results show a fairly promising fit to the PRA data, a slight over-prediction in reliability does exist. This observation, in addition to the inability of the system level approach to predict trends due to redundancy and engine out, leads to the identification of the subsystem level approach as more desirable.

The final conclusions from the example problem support the findings of Experiment 4. In Experiment 4 three options for applying the subsystem level reliability growth curves to the system fault tree were tested. From these tests it was concluded that option 1, anchor all subsystem curves at equivalent flight 0, is the most desirable due to its prediction accuracy. Since the example problem used actual PRA data from the STS vehicle, these three options were implemented in order to test their prediction accuracy. The results from each of the options were plotted above, which illustrated the shortcomings of options 2 and 3. These figures further support the conclusion that option 1 will achieve the highest prediction accuracy.

The STS example problem illustrated the utility of the CONTRAST method and verified its ability to accurately project launch vehicle reliability. A large scale test of the method is now required in order to confirm that the research objective has been successfully completed. This test problem will need to demonstrate the method on a relevant launch vehicle design problem, which will ultimately verify that the three requirements derived in Section 2.4 have been met.

CHAPTER V

APPLICATION & RESULTS

In Chapter 3 the CONTRAST method was developed starting from the observations from the literature review presented in Chapter 2. During the development of the method several research questions and hypotheses were developed, which required experimentation to accept or reject. These experiments have addressed all of the hypotheses; however, an additional experiment is required to test the CONTRAST method as a whole. Therefore, the purpose of this chapter will be to present a test problem that will demonstrate the method and verify that the original research objective has been met. This will be accomplished by applying the method to a real world example problem. The research objective and the derived requirements from Section 2.4 are restated below.

Research Objective: To formulate and implement a method that will quantitatively capture launch vehicle architecture effects on reliability and safety, in order to facilitate more informed decision making during early conceptual design.

Derived Requirements for Objective Completion:

1. The method shall produce quantitative estimates for reliability and/or safety of the given launch vehicle concepts

2. The method shall have sufficient accuracy to enable comparison between unique but similar concepts

3. The method shall be flexible enough to evaluate any potential launch vehicle concept within the defined architecture space

In order to test the completion of the research objective the CONTRAST method must be applied to a real world example launch vehicle problem. Therefore, the test problem will be set up based upon an actual launch vehicle, the Space Launch System (SLS). The SLS is currently being developed by NASA as the next generation heavy lift launch vehicle, which will enable manned exploration missions to the moon and beyond [124]. The SLS was chosen as a relevant example because design trade studies are currently being performed for future block upgrades of the vehicle. These upgrades will ultimately affect the vehicle's boosters and upper stage. Changes to these elements represent architecture options that will have an effect on the vehicle's reliability and safety. Thus, the SLS was deemed to be a perfect example vehicle for the application of the method.

To begin the test problem, a matrix of alternatives will be set up, which will represent the various architecture options that are available for an SLS-like vehicle. Additional options will be included in the matrix in order to capture more exotic design configurations. After generating the MOA the underlying reliability growth assumptions for each subsystem will be generated based upon literature or a parts count approach, which was discussed in Experiment 1.

Next, the automatic generation code must be set up for the example problem. During Experiment 4, automatic generation code was developed to produce both RBD and FTA equations from a matrix of alternatives. Based upon the results of that experiment a system level FTA will be used for the example problem. The code from Experiment 4 will therefore be used to setup an FTA for any combination of subsystems from the test problem matrix of alternatives.

After setup is complete, the CONTRAST method will be run for different vehicle architectures from the MOA. If the total number of architectures represented in the MOA is not prohibitive, all of the possible combinations will be run. If the total number is deemed too large, the combinations run through the method will be determined beforehand by assessment of the vehicle architectures that are most realistic.

When all of the runs have been completed an assessment of the CONTRAST method's performance can be done. This assessment will utilize the derived requirements for objective completion as stated above. First, the method must produce growth projections for all specified vehicle architectures. This requirement is expected to be easily completed because the growth models are able to produce projections for any given inputs.

The second and third requirements deal with the accuracy of the method and its ability to make comparisons between unique but similar vehicles. These requirements will be assessed via direct comparisons of the results of similar vehicles. The first proposed comparison will utilize a vehicle where the only change in architecture is the number of engines. Depending upon the number of engines an increase or decrease in reliability should be visible in the output. A second proposed comparison is the difference between utilizing redundant avionics or single string avionics. In this case an increase in reliability should be apparent when redundancy is utilized. Depending on the number of architectures that are run through the CONTRAST method, many different comparisons can be made. Ultimately, an exploration of the method output will determine the most appropriate comparisons. The successful completion of the second and third requirements will be supported using these comparisons.

The final test of the CONTRAST method will be to measure its required evaluation time. It is expected that the evaluation time will be fairly large due to the number of architectures represented in the MOA. If the evaluation time is very long, the analyst will not be able to analyze many of the architectures from the MOA. In this case, the method may not add any value to the design process. On the other hand, if the evaluation time is acceptable, many architectures can be run through the method. This will add a great deal of value to the design process as the analyst will be able to more fully explore the architecture space and evaluate design trades.

The runtime of the CONTRAST approach will be deemed acceptable if the completion of all runs requires on the order of days to complete. The benchmark that will be used for this metric is 64 hours, or two days and 16 hours. This is the length between close of business at 5 PM on a Friday and open of business, or 9 AM on Monday morning. It is considered acceptable to run the method over a weekend when the computer running the models may otherwise be idle.

Due to the combinatorial nature of the MOA, 100 percent of the architectures may not be evaluated during the 64 hour runtime. If this is the case, the percentage of the total architectures evaluated during that time period will be recorded. If this percentage is relatively low, a secondary acceptance criterion will be introduced. This secondary criterion will extend the allowable runtime to one week. As long as all of the desired output is produced, one week's worth of computer time is still considered acceptable for the method.

Based upon the tests proposed above, the fulfillment of the each of the derived requirements can be either confirmed or denied. The primary research objective will be considered satisfied if all of the derived requirements have been met. If this is the case, the CONTRAST method can be considered successful in providing a traceable approach for projection of launch vehicle reliability and safety during early conceptual design.

5.1 Definition of Alternatives

The first step in carrying out the test problem is to fully define the architecture space of interest. This will require the development of a matrix of alternatives, which contains all of the architecture options of interest. As discussed above, the SLS vehicle will be used as a model for setting up the alternatives. This vehicle was chosen because it is currently undergoing design activities related to the future block upgrades of its boosters and upper stage. Figure 38 gives a notional picture of the vehicle architecture. The SLS is a 2.5 stage vehicle, which has a liquid hydrogen and liquid oxygen core with a liquid hydrogen and liquid oxygen upper stage. The baseline SLS vehicle utilizes two solid rocket boosters, but advanced solid and liquid boosters have been proposed for future versions.

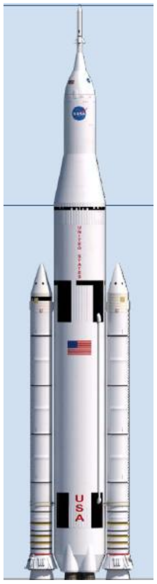


Figure 38: NASA Space Launch System

5.1.1 Matrix of Alternatives

From the generic description of an SLS-like vehicle, the rows of the test problem matrix of alternatives can be developed. These rows can be grouped into three categories; upper stage, boosters, and core. This section will develop the options contained within the rows of the test problem matrix of alternatives. Each of these options will be discussed in more detail in Section 5.2, which develops the reliability growth assumptions for each subsystem.

For the first category, the upper stage, five matrix rows can be defined. The first two rows pertain to the engine type and number of engines on the upper stage. Both the RL-10 and J-2X engines have been linked to the SLS program, making them two obvious options for engine type [151]. In addition, it will be assumed that a new engine development program may be possible for the upper stage. Due to the fact that the RL-10 is an expander cycle engine and the J-2X is a gas generator cycle engine, the new engine development will be assumed to be a staged combustion cycle. The addition of the new development staged combustion engine makes for three options in the upper stage engine type row. The RL-10 engine type, however, will be broken into two separate options in the matrix. These two options will represent an RL-10 with a fixed nozzle and an RL-10 with an extendable nozzle. Finally, the number of engines for the upper stage will be set between 2 and 5.

The remaining three rows pertain to the power system type, power system redundancy, and avionics system redundancy. The two redundancy rows will have two simple options; single or redundant. The redundant option will represent a fully redundant setup where there are two identical sets of hardware.

The power system type row will also contain two options. The first of these options is a standard battery driven power system, while the second is a new technology called integrated vehicle fluids (IVF). Integrated vehicle fluids was included in the matrix because it is a new technology that could have a large impact upon future vehicle capability. The IVF system, which integrates the power production and propellant thermal management, is currently being developed by ULA [177, 178]. For future upper stages this technology could have a profound impact upon the dry mass of the vehicle and ultimately its payload delivery capability [177].

The next category of rows within the test problem matrix of alternatives pertains to the vehicle boosters. Three rows of options can be identified, the first of which is the booster type. As discussed above, the baseline concept for the SLS vehicle will use a solid rocket booster that is derived from the STS program. For the next block upgrade of the SLS vehicle advanced solids or liquid boosters have also been proposed. The first three options in the booster type row will therefore be; STS SRB, advanced solid, and advanced liquid. An additional option will be added to this row in order to capture more exotic vehicle configurations. This option is a liquid fly-back booster. Although this type of booster has not flown to date, extensive studies have been performed, which include the evaluation of fly-back booster concepts for the Space Shuttle [8, 17, 140].

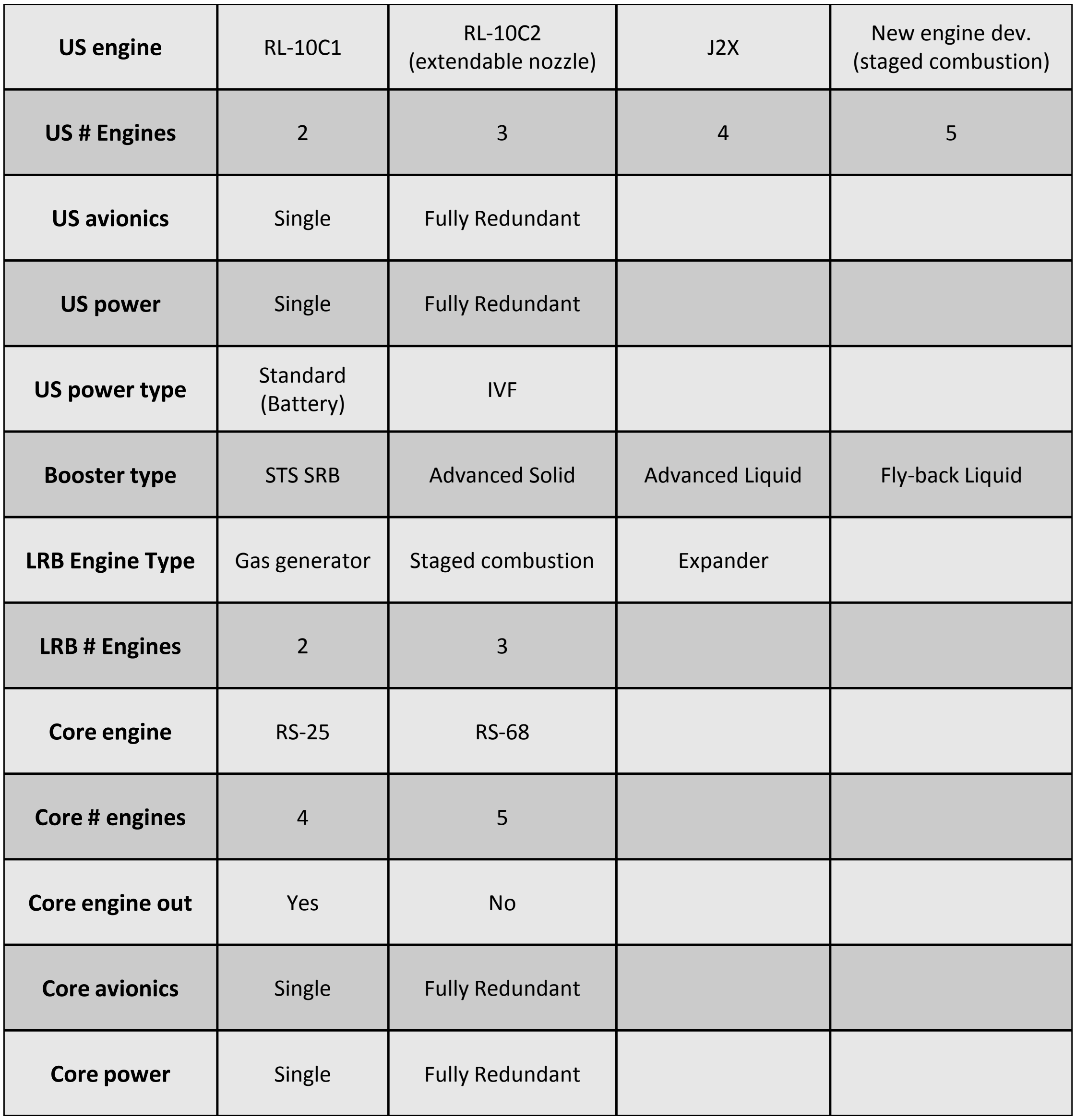
After the booster type row, two additional matrix rows are required specifically for the liquid boosters. These rows cover the liquid booster engine type and the number of engines per booster. The booster engine type row will contain three options representing the three primary engine cycle types; gas generator, staged combustion, and expander. These engine types were chosen to be generic in order to represent a new development program for the liquid booster engine. The assumption of a new development program for the booster engines is in line with the current developments of the SLS program. Although testing has been carried out on a historical engine, the F-1, the engine used on a future liquid booster will be produced using more modern design and production techniques [104]. With the identification of generic engine cycles for the booster engine type row, the number of engines per booster row options were set to 2 or 3. The number of booster engines assumptions are based upon the F-1 based and staged combustion engine based advanced boosters currently being considered for SLS [33, 34].

The final category of rows within the test problem matrix pertains to the core stage of the vehicle. This core stage burns in parallel with the two attached boosters and continues to burn after booster jettison. To represent the core stage configuration five rows will be used in the matrix of alternatives. The first two rows are engine type and core number of engines.

Based upon the design progress of the SLS vehicle, the options for the core engine type and number of engines are well defined. Early design studies performed for the SLS show two relevant engine options along with two number of engine options [86]. The options used for the core stage in the test problem will therefore be four or five RS-25 or RS-68 engines. In addition to the two engine number options an engine out row will be included. This row will allow for the modeling of an engine out capability in the core stage. Engine out was included because it is a primary approach for increasing vehicle reliability [88].

The remaining two rows for the core stage of the vehicle are identical to the avionics and power redundancy rows for the upper stage. These rows will include options for single or redundant power and avionics systems. As discussed above, the redundant option represents full redundancy of the specified subsystem.

After identifying all of the rows and options for the test problem, the final matrix of alternatives was assembled, which is shown in Figure 39. Section 5.1.2 will develop the assumptions associated with the test problem matrix of alternatives.



US engine	RL-10C1	RL-10C2 (extendable nozzle)	J2X	New engine dev. (staged combustion)
US # Engines	2	3	4	5
US avionics	Single	Fully Redundant		
US power	Single	Fully Redundant		
US power type	Standard (Battery)	IVF		
Booster type	STS SRB	Advanced Solid	Advanced Liquid	Fly-back Liquid
LRB Engine Type	Gas generator	Staged combustion	Expander	
LRB # Engines	2	3		
Core engine	RS-25	RS-68		
Core # engines	4	5		
Core engine out	Yes	No		
Core avionics	Single	Fully Redundant		
Core power	Single	Fully Redundant		

Figure 39: Test problem matrix of alternatives based upon the SLS vehicle

5.1.2 Matrix Compatibilities

The matrix of alternatives illustrated by Figure 39 defines the architecture space that will be analyzed for the test problem. Preliminary assumptions have been made in order to identify a reasonable number of options for each row in the matrix; however, additional assumptions are required. This section will outline the additional assumptions regarding in-compatibilities that will be enforced throughout the test problem.

The additional assumptions for the upper stage will be presented first.

The upper stage options within the matrix address the engine type, number of engines, power system type, avionics, and power redundancy. Within these options two primary compatibilities will be enforced. First, all of the number of engines options will not be compatible with all of the engine type options. This compatibility issue is related to the resulting thrust-to-weight ratio of the upper stage. If lower thrust engines are used, more engines are required to keep the stage in the desired thrust-to-weight range. On the other hand, if high thrust engines are used, fewer engines are required to meet the desired thrust-to-weight. A very low thrust-to-weight translates to a vehicle that will either reach orbit very inefficiently or fail to reach orbit altogether. A very high thrust-to-weight vehicle may require an excessively high structural mass to support its high acceleration. In addition, the engine mass for the high thrust-to-weight case will likely be very high.

Considering the options within the matrix of alternatives, the RL-10 engines are in a lower thrust class than the J-2X and the new staged combustion engine. For this reason, the RL-10 engines will be deemed compatible with all of the number of engines options. The high thrust engines, however, will not be compatible with all of these options. This is primarily true for the high number of engines cases where the thrust-to-weight of the vehicle would be excessively high. The resulting compatibility will enforce that the high thrust engines only be placed in a two engine configuration.

The second compatibility for the upper stage is related to the power system type and power system redundancy. In Section 5.1.1 two options for upper stage power system type were identified, standard, and integrated vehicle fluids. Due to the added complexity of the integrated vehicle fluids system it will be assumed that full redundancy cannot be achieved with this option. Therefore, only the standard power system is compatible with the fully redundant power system option.

Two more compatibilities regarding the boosters and core will be introduced for the test problem matrix of alternatives. The compatibility for the booster options is fairly obvious with the liquid engine type and number of engines rows only applicable when either advanced liquid or fly-back liquid have been selected in the booster type row. For the STS SRB and advanced solid options, the booster engine type and number of engines are not applicable.

The additional compatibility for the core stage pertains to the engine out options and the number of engines. As seen in the matrix, two options were included for core number of engines, 4 and 5. A simple compatibility assumption will be made that requires 5 engines in the core in order to enable engine out capability. The selection of 4 engines in the core stage will therefore only be compatible with the selection of no engine out capability.

Following the completion of the matrix of alternatives and in-compatibilities the reliability growth assumptions can be set up for the test problem. The following section will give a detailed description of the assumption set up for all of the subsystems that were identified in the matrix of alternatives. This section will first identify the subsystems that require reliability growth assumptions. Next, the specific subsystems will be discussed in more detail and the appropriate growth assumptions will be derived.

5.2 Reliability Growth Assumptions

From the matrix of alternatives in Figure 39, 20 individual subsystems that require reliability growth assumptions can be identified. The full list of subsystems includes 8 for the upper stage, 7 for the boosters, and 5 for the core. These subsystems have been broken out into five types: specific liquid engines, generic liquid engines, solid rocket boosters, avionics and power, and structures. Each of the subsystems within these five categories will be addressed in the subsequent sections.

Within these sections the assumptions for number of failure modes and probability of occurrence will be generated for all subsystems in the matrix of alternatives. The remaining reliability growth assumption, the fix effectiveness factors, will be approximated using the same distribution for all of the subsystems. Due to the fact that the FEF are the most difficult to accurately quantify, a single distribution will be used to ensure consistency between the individual subsystems.

As discussed in previous sections, the fix effectiveness factors are affected primarily by the management style and experience of the agency developing the launch vehicle. The vehicles produced by the test problem matrix of alternatives are assumed to have the exact same agency managing the program. Therefore, a standardized FEF distribution can be considered as a reasonable assumption for the test problem.

During the example problem in Section 4.2 a uniform distribution shape was decided upon for the FEF, which represents total uncertainty between a maximum and minimum value. The example problem utilized a fairly narrow range for the FEF values that was deemed appropriate for the manned STS vehicle. This distribution was uniform between 90% and 99%. Since the SLS vehicle will eventually be manned, the fix effectiveness factors for all of the subsystems in the test problem will also be modeled as a uniform distribution between 90% and 99%. This distribution represents a very focused failure reporting and correction scheme that is consistent with a manned vehicle program.

5.2.1 Specific Liquid Rocket Engines

The specific liquid rocket engines category pertains to pre-existing engines that were identified as options in the matrix of alternatives. These engines are considered to be currently operational, or in the case of the RS-25, recently retired. All of the engines in this category have been flown multiple times on operational launch vehicles or have undergone extensive developmental testing. Therefore, it is expected that each of the engine options will benefit from the availability of previous test or flight data.

Any data regarding reliability for these engines can be used to anchor the reliability growth assumptions.

5.2.1.1 RL-10 Engines Assumptions

The first specific engines from the matrix of alternatives are the RL-10 options for the upper stage. The RL-10 is a liquid hydrogen (LH2), liquid oxygen (LOX) burning engine that utilizes an expander cycle [143]. The RL-10 family of engines originated in the late 1950's when Pratt & Whitney began development of their first liquid rocket engine [134, 143]. The first static engine firing of the original RL-10 model was completed in 1959, making it the world's first LOX/LH2 rocket engine [154]. Since that time 10 operational models of the engine have been produced, which have supported upper stages of launch vehicles such as Atlas, Delta, and Titan [134].

Two options for the RL-10 were included in the matrix of alternatives in order to capture some of the variability between engine models within this family. The primary design characteristic that the matrix of alternatives made an effort to capture is the extendable nozzle feature that was added to later versions of the RL-10. This feature was added in order to increase both thrust and I_{sp} of the engine when operating at altitude [143]. It is therefore important to capture in the test problem matrix because it represents a possible trade between increased performance and increased complexity. Therefore the first option in the matrix, labeled RL-10C1, is assumed to have a fixed nozzle. The second option, the RL-10C2, represents the extendable nozzle version of the engine.

To begin the setup of the reliability growth assumptions for the RL-10 engines a schematic of the engine will be used to identify its key components. These components can be considered as the primary "failure modes" of the engine subsystem. Figure 40 below gives a basic schematic of the RL-10 engine. As seen in the figure, the RL-10 layout is relatively simple. The design consists of a turbine driven fuel pump, which is attached via gear transmission to the oxidizer pump. Regenerative cooling is used on the primary nozzle section and the fuel and oxidizer flow paths include flow control valves and a main fuel shutdown valve.

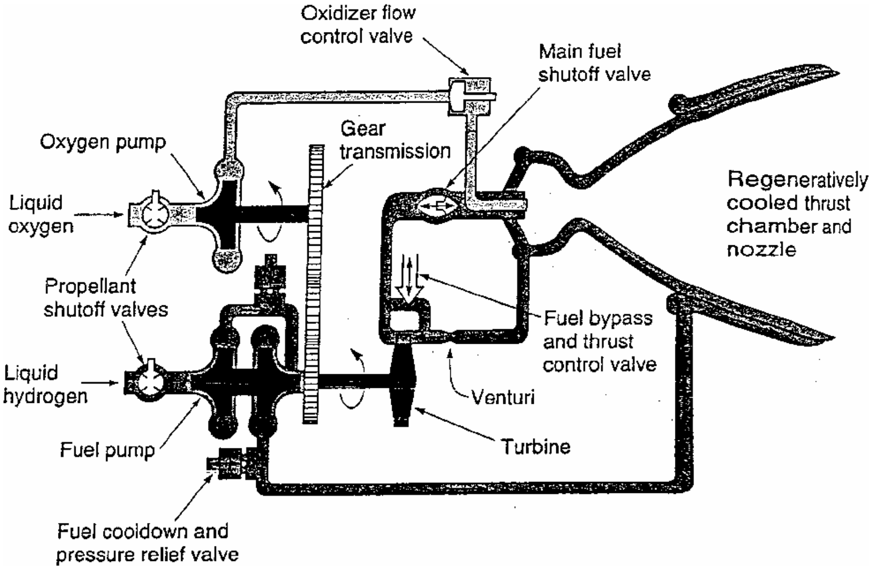


Figure 40: Simplified RL-10 schematic [154]

From Figure 40 seven primary components can be identified as assumed failure modes for the RL-10 reliability growth model. These components include the fuel pump, oxidizer pump, gear transmission, feed control, combustion chamber, nozzle, and the regenerative cooling heat exchange. These modes can be used for the first RL-10 option in the test problem matrix of alternatives; however, additional modes will need to be introduced for the extended nozzle version. Figure 41 illustrates the RL-10 engine with its nozzle extension in the stowed position.

The stowed nozzle configuration illustrates three additional components that are candidates for inclusion in the list of failure modes. The first additional component is the extension system, which uses a motor to drive a pulley system that controls the nozzle actuation [132, 143]. Upon extension of the nozzle, the second key component is the latching joint between the fixed nozzle section and the movable nozzle section. This latch joint must close properly in order to ensure desired nozzle performance. Finally, an additional failure mode can be added for the movable nozzle section itself.

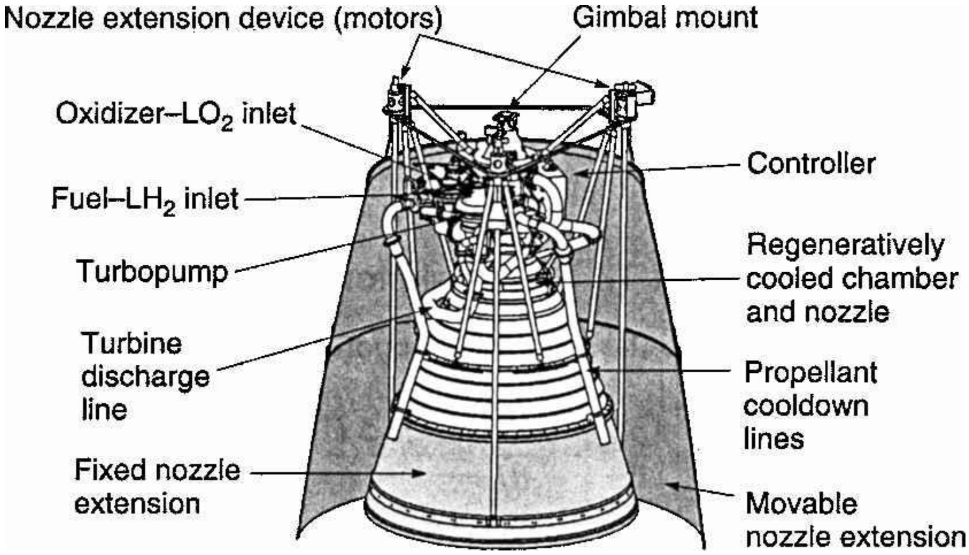


Figure 41: RL-10 engine with stowed nozzle extension [154]

Using the basic RL-10 schematic in Figure 40 and the stowed nozzle picture in Figure 41 the number of failure modes for the two engine options have been estimated to be 7 and 10. These assumptions were derived using the basic parts count type approach which was demonstrated during the example problem in Section 4.2. The next required assumption for the RL-10 engine is the probability of occurrence distribution for the failure modes. This distribution can be derived using a reliability estimate from the RL-10 flight history.

In reference [71], Go discusses the flight history of upper stages that have utilized the RL-10 engine between 1962 and 2005. In all, this flight history represents 190 vehicle configurations with RL-10 engines [71]. Out of 190 flights only 22 failures occurred, of which 12 were attributed to the upper stage [71]. Go lists each of these upper stage failures and their effects, which illustrates that only 3 of the 12 upper stage failures stemmed from the RL-10 engine. Two of these engine failures were due to turbopump failure caused by material contamination, while the third occurred due to a boost pump failure [134].

In very simple terms, this flight data suggests that the RL-10 engine has demonstrated a probability of success of 187/190 or 0.9842. However, since these early engine failures the RL-10 has achieved a 100% mission success rate [134]. Therefore, the initial probability of failure of the RL-10 for the purpose of the test problem will assume a mean reliability of 0.9842 with a standard deviation of 0.016, which allows for the maximum reliability to approach 1.0 as shown by the mission success rate.

It is important to note that this reliability estimate is for the RL-10 engine as a whole. However, the probability of occurrence values are needed for the failure modes of the engine, which are one level of characterization lower. Therefore, an additional step is needed in order to derive the probability of occurrence distribution for the assumed RL-10 failure modes.

The process used to derive this distribution is described in more detail in Appendix C. Approach 2 in Appendix C was used for the RL-10 probability of occurrence distribution because the engine level mean reliability and standard deviation are known. The resulting probability of occurrence distribution for the RL-10 failure modes is Beta(0.078, 33.9). This distribution will be used for both of the RL-10 options in the matrix of alternatives because it is representative of the expected reliability of the entire family of engines.

5.2.1.2 J-2X Engine Assumptions

The next specific liquid engine from the matrix of alternatives is the J-2X upper stage engine. The J-2X is a liquid oxygen, liquid hydrogen burning gas generator cycle engine [119]. The engine is derived from the flight proven, Apollo era, J-2 rocket engine as well as the experimental J-2S engine [119]. Various components within the J-2X design can also be traced to the RS-68 engine, which will be discussed later [119]. Figure 42 below shows a basic diagram of the original J-2 engine, which shares the same components as the J-2X.

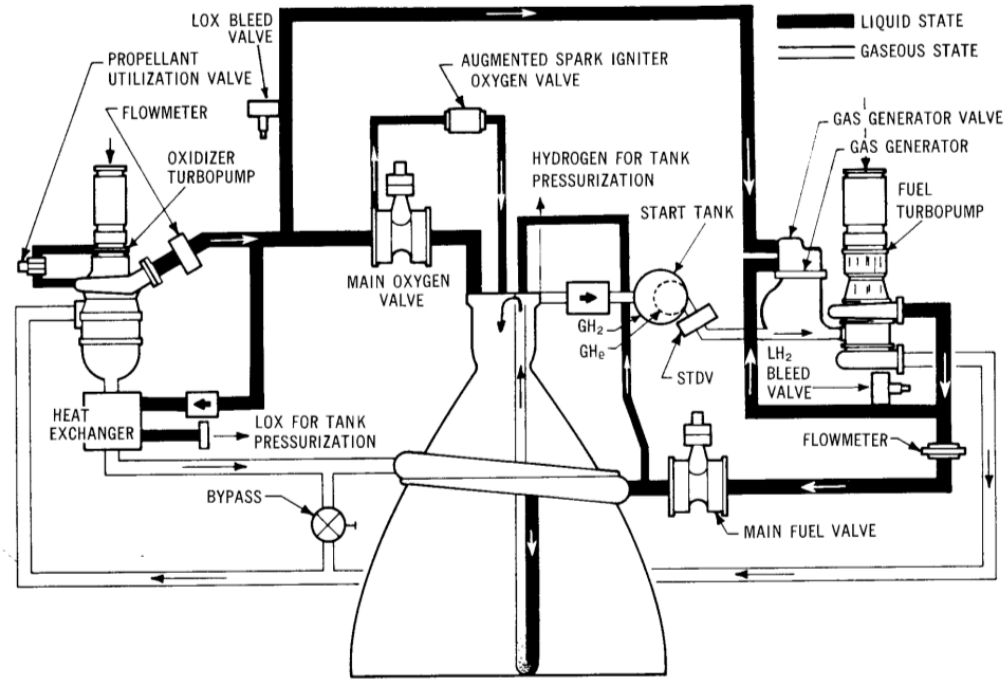


Figure 42: Simplified J-2 engine schematic [110]

Similar to the process taken with the RL-10 engine a basic parts count approach will be used to identify the number of "failure modes" for the J-2X engine. As seen in the figure, 7 primary components can be identified. These components include the fuel and oxidizer turbopumps, heat exchanger, gas generator, feed control system, main combustion chamber, and nozzle. The resulting parts count is the same as what was seen for the first RL-10 option in the previous section. However, the J-2X engine will still be differentiated from the RL-10 due to changes to the probability of occurrence distribution.

The probability of occurrence distribution for the J-2X engine can be derived using reliability data from the J-2 engine as well as the test history of the J-2X engine and its components. As mentioned previously, the J-2 engine was flown on the Saturn V launch vehicle during the Apollo program. During this time the J-2 engine was flown 11 times on the Saturn S-IVB third stage and 10 times on the S-II second stage [127]. The Apollo era flight history shows that in total, 61 J-2 engines were flown without any loss of mission type failures.

McFadden presents analysis of the Apollo era flight and test data for the J-2 engine in reference [103]. Within this reference a database of engine data was collected and the mean reliability of various liquid rocket engines was estimated. In addition to the mean reliability estimates, McFadden also includes confidence bounds in the form of 5th and 95th percentile values. The J-2 reliability values given for the mean and percentiles are 0.9916, 0.9697, and 0.9252 [103]. These values can be used to directly estimate a reliability distribution for the J-2 engine, however, the more recent testing of the J-2X engine should be considered first.

Testing of J-2X engine components began in 2006 with sub-scale hot fire testing of the main injector [20]. Component testing continued on until the first system testing began in February 2008, which lead to the successful completion of 6 power pack assembly tests [20]. The summer of 2011 marked the first full scale test of the development engine [119]. During the full scale testing four engine test articles were successfully hot fired for a cumulative duration of five hours [137]. Only one pre-mature shutdown event occurred during the testing of these engines [9].

Overall, the successes of the J-2X test program point to the engine being as reliable as or more reliable than the original J-2. This is acknowledged by Buzzell, who notes that the loss of mission reliability is expected to be an order of magnitude higher than the heritage engine [20]. Considering this comment, an order of magnitude increase from the mean reliability value given by McFadden for the J-2 would put the J-2X reliability estimate near the highest percentile value of 0.9916. Since this value lies on the upper portion of the data given by McFadden, the J-2 reliability data can be used as a conservative starting point for the J-2X reliability.

To complete the probability of occurrence assumptions for the J-2X engine it was assumed that the reliability distribution given by McFadden in reference [103] is a logical starting point for the engine reliability. Since this reliability data gives both the mean and percentile values an alternative approach to what was done for the RL-10 can be used to derive the probability of occurrence distribution for the failure modes. The details of this approach are discussed in Appendix C. Once the engine level reliability distribution is available, the probability of occurrence distribution for the failure modes can be estimated. The resulting probability of occurrence distribution is Beta(0.24,54), which will be used in the reliability growth model for the J-2X engine.

5.2.1.3 RS-25 Engine Assumptions

The next liquid rocket engine is the RS-25, which is also known as the Space Shuttle Main Engine. This engine is the first option for the core stage engine type row of the matrix of alternatives. The RS-25 engine is a liquid oxygen, liquid hydrogen burning staged combustion engine [131, 165]. The engine was developed in the 1970's by Rocketdyne for use on the Space Transportation System [154, 165]. Since its inception, the RS-25 has benefited from an extensive test and flight history including over 1 million seconds of cumulative test time and 135 successful operational flights [11, 165]. Figure 43 provides an illustration of the RS-25 engine layout.

As discussed in Section 4.2.2 of the STS example problem, a parts count type approach can be used to identify the number of failure modes for the RS-25. However, an alternative approach will be applied for the test problem due to the extensive history of the RS-25 engine. The extensive history of the engine means that ample design, test, and operational data are available for estimating reliability. The number of failure modes assumption for the RS-25 will therefore be made based upon FMEA data from the development and early operational stages of the engine. A similar statement could be made about the RL-10 engine; however, RL-10 design data such as FMEA worksheets are not publicly available as is the case for the RS-25.

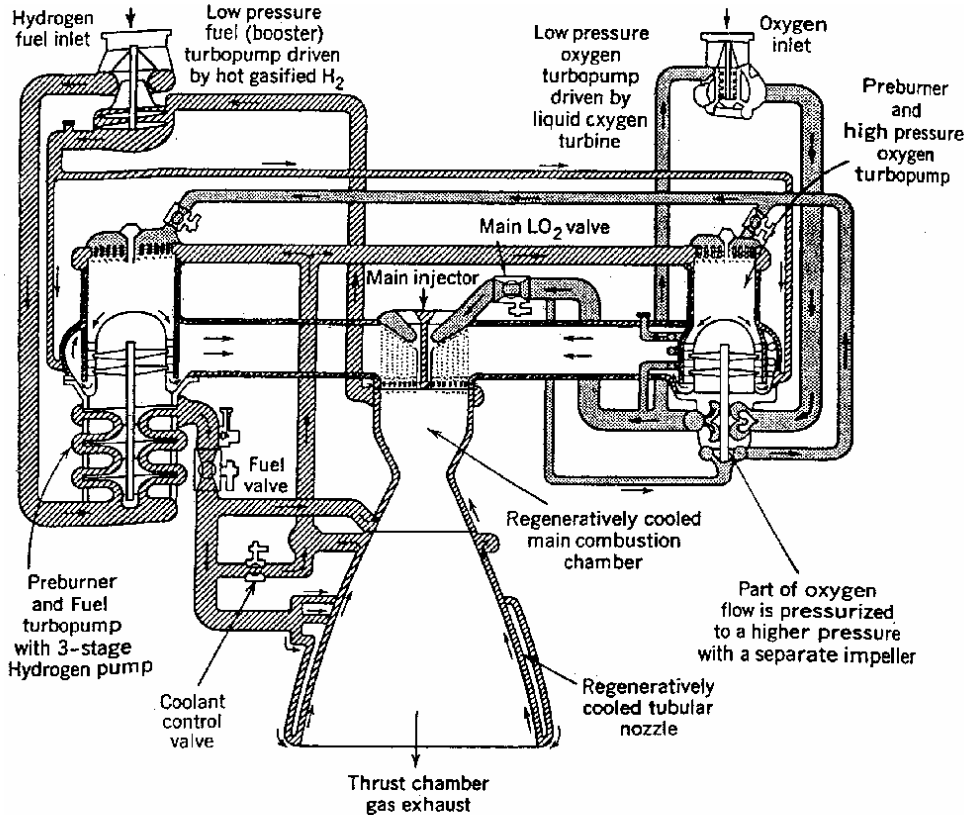

Hydrogen fuel inlet
Low pressure fuel (booster) turbopump driven by hot gasified H_2
Low pressure oxygen turbopump driven by liquid cxygen turbine
Oxygen inlet
Preburner and high pressure oxygen turbopump
Main LO_2 valve
Main injector
Fuel valve
Regeneratively cooled main combustion chamber
Part of oxygen flow is pressurized to a higher pressure with a separate impeller
Preburner and Fuel turbopump with 3-stage Hydrogen pump
Coolant control valve
Regeneratively cooled tubular nozzle
Thrust chamber gas exhaust

Figure 43: Simplified RS-25 engine schematic [154]

The first data source that can be used for failure mode estimation is reference [70], which is a technical report from early in the Space Shuttle program that presents an SSME failure data review and diagnostic survey. Within this report, specific failure modes of the SSME are identified and characterized based upon their severity and likelihood of occurrence. This report also discusses the effects of the various failure modes based upon their estimated cost and time between initiation of the mode and engine loss or shutdown. Using these factors an overall risk value is assigned to each failure mode ranging from loss of vehicle at 1.0 down to part is "OK" at 0.1. This reference is by far the most detailed and most well suited for generating the reliability growth assumptions for the RS-25 engine.

Two additional data sources can be used in tandem with reference [70]. In reference [83] a review of the Independent Orbiter Assessment (IOA) FMEA/CIL is presented, which gives the number of critical items for each of the major subsystems in the Space Shuttle Orbiter. This overview gives a detailed list of the original critical items that were identified prior to the IOA study and the remaining critical items that were accepted at the conclusion of the study.

Another additional source, reference [111], also addresses the results of the IOA study FMEA/CIL. This reference is a report presented to the congressional committee on science, space, and technology, which requested an additional safety assessment of the SSME. The report gives an extensive review of FMEA/CIL analyses, reliability assessments, and proposed improvements to the RS-25.

From the three references above the number of failure modes for the RS-25 can be easily estimated. First, the generic critical items list from reference [83] suggests that the main propulsion system (MPS) contains 43 accepted critical items. This estimate would suggest a maximum of 13 critical items per engine. However, the MPS includes all three engines as well as the fuel and oxidizer cross-feed systems, which means the critical items specific to the RS-25 may be fewer.

This observation is supported by the hazard analysis presented in the second source from above, reference [111]. In this report a hazard analysis fault tree is shown, which identifies 16 total SSME failure modes that could result in a loss of crew and/or loss of vehicle. It also states that seven of the identified failure modes are considered to be controlled through inspection and testing. Therefore, nine of the sixteen failure modes are labeled as accepted risks for the SSME. Note that this number represents nine accepted failure modes per engine.

When considering the first source, reference [70], this failure mode estimate is further supported. Within the analysis in reference [70] a total of 190 failure modes for the SSME are identified and categorized. A majority of these modes, however, fall in the lower risk factor categories such as minor local damage or piece part damage. The categories of interest are the top most levels, which include loss of vehicle, probable loss of vehicle, and loss of engine. For these top three levels, 3 modes are placed in the loss of vehicle category, 7 modes fall in the probable loss of vehicle category, and 3 modes exist in the loss of engine category.

These results are in line with the previous sources, showing around 10 failure modes per engine that can lead directly to a loss of vehicle event. It is also interesting to note that the simplified parts count approach results in approximately 10 failure modes for the RS-25. This estimate includes the major components such as; 2 fuel pumps, 2 oxidizer pumps, 2 pre-burners, heat exchanger, combustion chamber, flow control, and nozzle. Therefore, 10 failure modes for the RS-25 engine will be assumed for the test problem.

After generating the estimate for number of failure modes, the probability of occurrence for these modes is required. To estimate the probability of occurrence distribution for the RS-25 a similar approach to what was used for the J-2X will be applied. Within reference [103] a database of liquid engine tests and operational flights is analyzed. From the data a mean reliability as well as 5th and 95th percentiles are derived for many different liquid engines including the SSME. From this reference the engine level reliability for the RS-25 will be assumed to have a mean of 0.9885 and percentiles of 0.9478 and 0.9994. This data will define the initial reliability of the RS-25 in the test problem. From the mean and percentiles the probability of occurrence distribution for the failure modes was estimated using the third approach in Appendix C. The resulting probability of occurrence distribution for the RS-25 failure modes is Beta(0.09,54.5).

5.2.1.4 RS-68 Engine Assumptions

The final specific liquid engine for the test problem is the RS-68, which is used in the core stage of the Delta-IV launch vehicle [155, 161]. The RS-68 is a gas generator cycle engine that utilizes liquid oxygen and liquid hydrogen [81, 174]. Development of the engine by Rocketdyne commenced in 1997, which resulted in certification for use on the Delta IV in 2001 [175]. A great deal of the eventual RS-68 design stemmed from earlier NASA Space Transportation Main Engine conceptual studies, which focused on reducing the cost and development time required to produce a new liquid engine [175]. A basic schematic of the engine is shown in Figure 44 below.

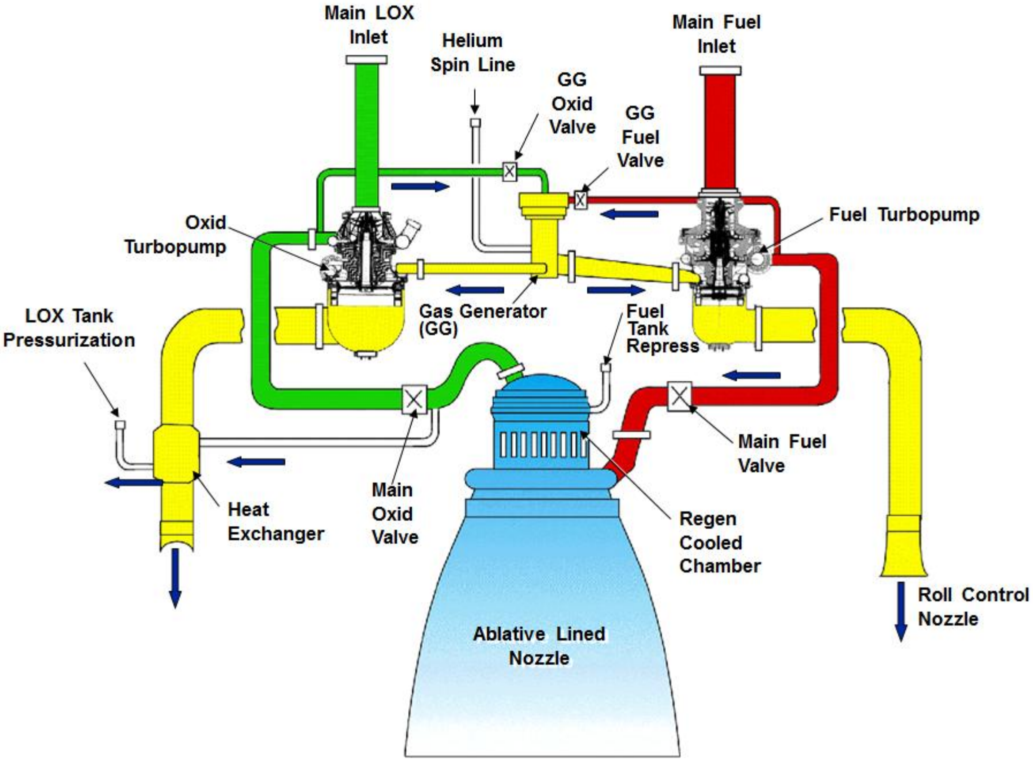


Figure 44: Simplified RS-68 engine schematic [175]

Since the RS-68 was first certified for flight in 2001, its flight history is not nearly as extensive as the other liquid engines that have been discussed. In addition, as was the case with the RL-10, the RS-68 engine is still in production. This means any detailed data regarding failure modes and effects is not publicly available. Therefore, the simplified parts count approach will be used to generate the number of failure modes assumption for the RS-68.

As can be seen in Figure 44, the RS-68 contains one oxidizer and one fuel turbopump. Both pumps are connected to a gas generator that is centrally located on the diagram. The oxidizer pump is connected to the heat exchanger, which then connects to the LOX tank pressurization system. An interesting feature of the RS-68 is the propellant flow from the fuel turbopump. Some of the flow from this pump is fed directly to a secondary nozzle, which is used for roll control. In addition to the pumps and gas generator, the RS-68 contains components for flow control as well as a primary combustion chamber and nozzle. In all, the parts count for this engine yields an estimate of 8 failure modes.

Now that the number of failure modes have been estimated, the probability of occurrence for these modes must be determined. For the RS-68 the test and flight histories can be used to generate an initial reliability distribution for the engine. From this distribution the probability of occurrence values will be determined.

The RS-68 is considered to be the first liquid rocket engine that was fully designed using computer-aided design and analysis programs [155]. Due to the extensive use of computer-aided design as well as new manufacturing approaches, the length of the flight certification test program was decreased significantly compared to previous engines [155, 175]. In all, only 183 tests were performed for flight certification, which is a factor of four smaller than the number of tests required of the SSME [175]. Through these tests no major failures were encountered and less than 20 pre-mature test cut-offs were required due to engine anomalies [175]. In addition, the RS-68 has been flown on 27 total Delta IV flights with no loss of mission failures [89].

The flight and test history of the RS-68 show a near perfect operational record for the engine. Although it has demonstrated a near 100% reliability, the predicted reliability of the engine is more appropriate for use in the test problem. This is primarily due to the fact that the engine has only flown 27 operational flights, which is a small number compared to the RL-10 or SSME. Wood notes that reliability predictions for the RS-68 have been carried out using comparative design assessments that take into account parts count, complexity, fabrication, inspection, and operating environments [175]. From this assessment comparisons were drawn to detailed historical data from the SSME, which resulted in a predicted reliability of 0.9987 for the RS-68 [175].

For the test problem a mean reliability of 0.9987 will be assumed for the RS-68. Similar to the RL-10 engine, the standard deviation for the RS-68 will be assumed as 0.0013, which will allow the maximum reliability to approach 1.0 as demonstrated by the flight history. As with all of the other specific liquid engines, the probability of occurrence distribution for the RS-68 failure modes was derived using the procedures described in Appendix C. Since the mean and standard deviation of the engine reliability were defined, the second approach from Appendix C was used. This derivation resulted in a probability of occurrence distribution of Beta(0.0759,36.02), which will be used for the RS-68 in the test problem.

5.2.2 Generic Liquid Rocket Engines

After developing the assumptions for the specific liquid rocket engines from the matrix of alternatives, the generic engines can be assessed. These generic engines represent three different cycle types; gas generator, staged combustion, and expander. All three of these engines will be options for the advanced liquid booster and the fly-back booster. The generic staged combustion engine will also be an option for the upper stage engine type. The assumptions will be setup based upon a parts count from a generic description of the engine layout.

5.2.2.1 Generic Gas Generator Cycle Assumptions

The first engine cycle type to consider is the gas generator. This engine type will be an option for the advanced liquid booster and the fly-back liquid booster from the matrix of alternatives. The advanced booster utilizing a gas generator engine represents a booster architecture that is currently being considered for the SLS vehicle [33, 34]. The gas generator engine for this concept is based upon the F-1 engine from the Apollo era [33].

The gas generator cycle is one of the most common engine cycles due to its relative simplicity[156]. Gas generator engines typically have lower operating pressures, smaller inert masses, and lower development costs [156]. Although this cycle provides multiple benefits, it generally supplies less performance with a specific impulse a few percent lower than other cycles [156]. Figure 45 below gives a simplified diagram of a gas generator cycle engine.

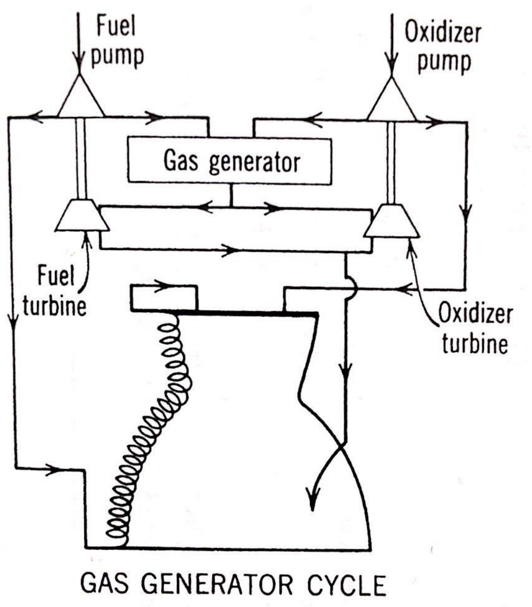


Figure 45: Simplified gas generator cycle engine schematic [156]

As seen in the figure, the engine cycle utilizes a gas generator to drive the turbine(s), which are attached to the fuel and oxidizer turbopumps. Gas generator engines can either be open or closed cycle depending upon the path of the turbine exhaust gases. An open cycle engine will dump this gas overboard through a low area ratio nozzle, while a closed cycle engine will aspirate the exhaust gases into the diverging section of the nozzle [156]. The latter option is shown in Figure 45.

From the basic schematic the primary components of a generic gas generator can be identified. These components include the fuel turbopump, oxidizer turbopump, turbine, gas generator, feed control, heat exchanger, combustion chamber, and nozzle. The simple parts count approach therefore yields an estimate of 8 primary failure modes for the generic gas generator.

After estimating the number of modes for a gas generator engine, the generic probability of occurrence distribution must be determined. Due to the fact that a generic engine is being considered, there is no reliability data to base the probability of occurrence values on. Therefore, the generic probability of occurrence distribution for a complex system from Experiment 1 and 2 will be applied. This distribution, Beta(0.22,8.75), was demonstrated within both experiments as well as the example problem in Section 4.2. The results of the example problem illustrated that this generic probability of occurrence distribution produced satisfactory results in terms of prediction accuracy. Thus, the generic distribution will be used for all of the generic liquid rocket engines as well as any subsystems that do not have reliability data for comparison.

5.2.2.2 Generic Staged Combustion Cycle Assumptions

The next engine type to consider is the staged combustion cycle. This generic engine will be included in the options for the advanced liquid booster, fly-back liquid booster, and upper stage engine type. It will be assumed that the generic layout of the engine will not differ between the two applications of the engine.

The staged combustion cycle is a closed cycle that is more complex than a gas generator [156]. It operates with a higher turbine flow and requires higher pump discharge pressures to overcome the extra pressure drop due to the pre-burner [156]. Staged combustion engines also tend to operate at much higher chamber pressure because the turbine exhaust flow is injected into the main combustion chamber [156]. Although staged combustion engines tend to be heavier and more complex, they do provide a high specific impulse [156]. A basic schematic of a staged combustion engine is shown in Figure 46.

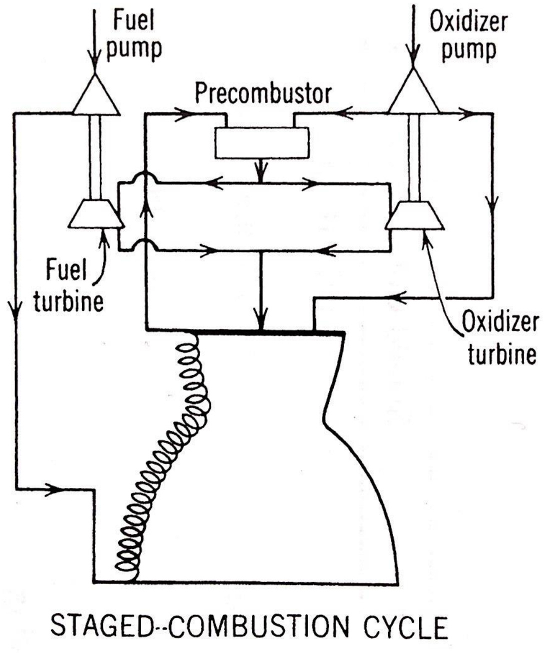


Figure 46: Simplified staged combustion cycle engine schematic [156]

The staged combustion cycle utilizes a coolant flow in which the liquid fuel is fed through a cooling jacket around the engine nozzle and combustion chamber [156]. After heating, the fuel is fed to the pre-combustor, which burns all of the fuel along with part of the oxidizer. The high-energy gas from the pre-combustor is used to drive the turbines for the turbopumps [156]. Ultimately, the exhaust gas from the turbines is injected into the combustion chamber where it is burned with the remainder of the oxidizer [156].

For a generic staged combustion engine the simple parts count approach yields eight primary components. These components include the fuel turbopump, oxidizer turbopump, pre-combustor, turbines, heat exchanger, nozzle, and combustion chamber. It is important to note that multiple pre-combustors and multiple turbopumps may be used in a staged combustion engine. For example, the SSME contains two pre-burner chambers as well as two pumps for each of the fuel and the oxidizer [156]. To account for the possibility of multiple pumps or multiple pre-combustors, the assumed number of failure modes for the generic staged combustion engine will be increase to ten.

As explained in the previous section, the generic liquid engines do not have previous reliability data to draw from in order to generate the probability of occurrence assumptions. For this reason, the generic probability of occurrence distribution from the experiments section will be used for all of the generic subsystems. This distribution, Beta(0.22,8.75), is meant to represent a “complex system” and was shown to produce acceptable results in terms of prediction accuracy.

5.2.2.3 Generic Expander Cycle Assumptions

The final generic engine is the expander cycle liquid rocket engine. This engine option will be available for the advanced liquid and fly-back booster options in the matrix of alternatives.

Similar to staged combustion, the expander cycle is a closed cycle [156]. The expander cycle also uses a cooling jacket to supply energy to the fuel flow in order to drive the turbopumps and regeneratively cool the engine nozzle and combustion chamber [156]. This cycle is different from staged combustion in that it does not utilize a pre-burner. Therefore, expander cycle engines are relatively simple and have a low engine mass [156]. These engines also provide higher performance than a gas generator cycle [156]. Figure 47 below gives a simplified layout of an expander cycle engine.

In the expander cycle the fuel and oxidizer are fed through their respective turbopumps. These turbopumps are driven by the coolant flow of fuel through the nozzle cooling jacket [156]. The turbine exhaust and the oxidizer flow are then injected into the combustion chamber. All of the propellants in an expander cycle engine are fully burned in the combustion chamber and expanded efficiently in the exhaust nozzle [156].

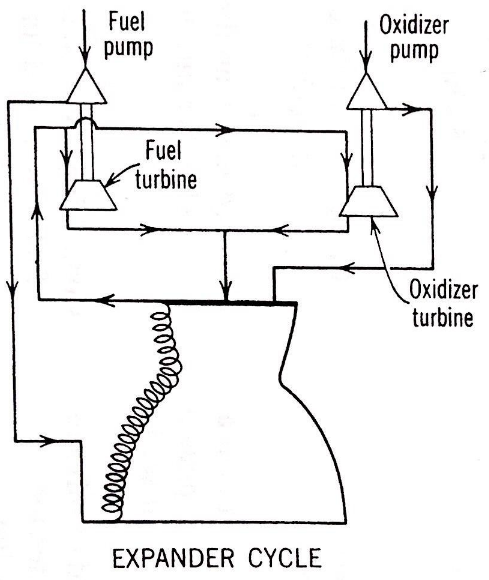


Figure 47: Simplified expander cycle engine schematic [156]

From the simplified layout of the expander cycle engine a parts count of 7 can be obtained. This parts count includes the fuel turbopump, oxidizer turbopump, turbine, feed control, heat exchanger, nozzle, and combustion chamber. As with the other generic engines the simple parts count failure mode assumptions will be accompanied by a generic probability of occurrence distribution. This distribution is Beta(0.22,8.75), which was implemented successfully in the example problem in Section 4.2.

5.2.3 Solid Rocket Boosters

The next subsystems to address from the test problem matrix of alternatives are the solid rocket boosters. Two options for the solid boosters were included in the matrix; the STS SRB and advanced solid. The first option will be based upon the reusable solid booster that was implemented for the Space Shuttle. The second option is a proposed future upgrade to the shuttle solid booster that aims to reduce the booster mass in order to improve performance.

The STS SRB was described in more detail during the example problem in Section 4.2. In this section the number of failure modes for the STS SRB was estimated to be 7 for the example problem. However, the SRB for the test problem will have more failure modes. Although the SLS SRB is derived from the shuttle SRB, the baseline design contains an additional segment [3]. The new derivative of the booster will also produce more thrust, while using new technologies and materials to reduce mass [3].

To generate the number of failure modes assumption for the STS derived SRB a review of its basic components is required. As noted above, the booster will consist of 5 fuel segments. In addition to these segments an igniter is placed atop the booster and the nozzle assembly lies on the bottom. The nozzle assembly also contains mechanisms for thrust vector control and launch pad tie down for the entire vehicle. The booster nose cone above the igniter segment typically houses the avionics package. The solid rocket booster components also include the primary attach points between the core of the vehicle and the booster itself. The forward attach point is the primary load bearing structure, while an aft attach point may be used for stability. From the generic description of the components within the solid rocket booster an estimate of 12 failure modes will be used for the test problem.

The second solid booster to consider is the advanced solid. This booster is a proposed upgrade to the baseline STS derived booster. The advanced booster will implement new technologies, such as composite casings, to improve upon the performance of the baseline booster. The design for the advanced solid has not been determined to this point, however a concept has been illustrated by ATK that contains four segments with a redesigned composite casting and nose cone [10]. Since the state of the advanced booster design is still in flux the test problem will consider it to have the same number of components as the STS derived SRB. The differentiating factor between the two boosters will therefore be the initial reliability of each concept. The probability of occurrence for the STS derived SRB will be estimated based upon the historical reliability of the STS SRB. The advanced booster will be treated as a generic complex system, which will use the same probability of occurrence distribution as was used for the generic liquid rocket engines.

In total the STS SRB flew 135 missions with only 1 major failure on STS-51L [27]. In addition, 52 full scale tests of the booster were performed in support of the Space Shuttle program [107]. This test and flight history suggests that 322 boosters have operated with only 1 major failure. A ratio of 1 failure in 322 trials gives a demonstrated reliability of 0.9968. Considering only the operational flights, the demonstrated reliability of the SRB is 0.9962. Therefore, the initial reliability of the STS derived SRB will be assumed at 0.9962 for the test problem. The procedures for deriving the probability of occurrence distribution from this value are presented in Appendix C. The results from the second approach in the Appendix C gives a probability of occurrence distribution of Beta(0.004,11.4) for the STS SRB.

5.2.4 Avionics and Power

The next subsystems from the test problem matrix of alternatives are the avionics and power systems. These subsystems will be included on the core and upper stage of the vehicle. Note that the solid and liquid boosters also typically contain avionics of their own, however, it was decided that the core and upper stage avionics were more critical to a successful launch. The booster avionics are therefore secondary to the main flight computers on the core and upper stage.

Both the core and upper stage avionics will utilize the same probability of occurrence distribution for their respective failure modes. This distribution has been discussed multiple times before and represents the probabilities of occurrence for a complex system. The core and upper stage avionics subsystems will use Beta(0.22,8.75) for the probability of occurrence.

The number of failure modes assumption for the core and upper stage avionics, however, will differ slightly. A difference in number of failure modes was implemented in order to capture the higher complexity that is expected of the core avionics system.

This complexity is due to the core avionics handling the guidance, navigation, and control of the vehicle for a majority of the trajectory. In addition, the core avionics will communicate with the booster avionics and control the staging events such as booster jettison and core/upper stage separation.

To generate the number of failure modes assumption for the core avionics, FMEA data from the STS can be used. This data will be used because the functions of the Shuttle avionics system during launch are considered to be similar to that of the vehicle in the test problem. During ascent the orbiter avionics handle the guidance, navigation, and control of the vehicle, which has two boosters and three liquid engines. It also handles the throttling and staging events throughout the trajectory. The core avionics system in the test problem will have similar functions, handling the guidance, navigation, and control for the vehicle with two boosters and four or five liquid engines. The core avionics will also handle two staging events, which is similar to the orbiter avionics that handle booster separation and external tank jettison.

During the late 1980's an independent study of the orbiter FMEA/CIL analyses was performed by McDonnell Douglas. From this study, three reports regarding the orbiter avionics were produced, which are applicable to the test problem core stage avionics assumptions. The first of these reports, produced in 1986, presents a detailed look at the failure modes and critical items within the data processing system of the orbiter [97].

Within the report in reference [97], seven primary components within the data processing system are assessed. From this assessment a list of 78 total failure modes from the FMEA are identified along with 23 critical items. The 78 identified failure modes were then classified into 6 failure criticality levels. The first of these levels is of most interest for the example problem because it represents a loss of crew or loss of vehicle event [97]. Two of the 78 total failure modes are classified in this level for the data processing system. Therefore, this portion of the avionics subsystem will be assumed to contribute 2 failure modes.

The next report on the orbiter avionics addresses the guidance, navigation, and control portion of the orbiter avionics. This report assesses the GNC system in three individual segments; major components, function switches, and power switches [158]. In total, the GNC study identified 175 failure modes and 36 critical items. Similar to the previous study, six criticality levels were used to divide these failure modes. The resulting criticality assessment resulted in 8 failure modes being classified as causing loss of crew or vehicle. Therefore, 8 modes can be assumed for the GNC portion of the avionics system.

The third report from the orbiter FMEA/CIL assessment addresses the instrumentation segment of the avionics system. This report divides the instrumentation system into 11 individual categories of equipment [67]. From all of these categories the total number of failure modes was identified as 107 with 22 critical items [67]. This report also uses the same 6 level criticality assessment as discussed above. For the instrumentation system none of the failure modes were assigned to the first criticality level. Since none of the failure modes were classified in the top criticality level, the second level failure modes will be added to the assumptions for the test problem. This will give a more conservative estimate for the avionics failure modes. Therefore, two additional modes will be contributed to the total number of avionics subsystem modes.

The review of the three assessments of the orbiter avionics returned an estimate of 12 failure modes that would directly cause a loss of crew or loss of vehicle event. In order to make the test problem assumptions for the core avionics more conservative, this number of failure modes will be increased to 15. This increase was implemented because the core stage avionics will control the same number of boosters as the STS but more liquid rocket engines. In addition, the avionics system for the core stage will operate in a different environment than the orbiter avionics, which could contribute more failure modes.

As discussed previously, the upper stage avionics will be assumed to have fewer failure modes than the core avionics. This is due to the expected reduction in complexity of the upper stage avionics. A reduction in complexity is expected because the upper stage avionics will handle fewer staging events, fewer engines, and a much smaller instrumentation system than the core. From the orbiter avionics assessment 12 failure modes were identified that would cause a loss of crew or loss of vehicle. A reduction in complexity warrants a decrease in the number of assumed failure modes for the upper stage avionics. In order to preserve a relatively conservative estimate, the number of failure modes for the upper stage avionics will be reduced to 10. Therefore, the core stage avionics will be assumed to contain 5 more failure modes than the upper stage to account for additional engines, separation events, and instrumentation.

After developing the assumptions for the generic avionics subsystems in the core and upper stage, the power subsystems can be considered. Similar to the avionics, the core power subsystem number of failure mode assumption will be developed based upon STS FMEA data. The upper stage power system will be developed based upon a generic parts count approach. This parts count approach will be used for both the standard and integrated vehicle fluids options in the matrix of alternatives. The probability of occurrence distribution will again be assumed as a generic complex system, which is Beta(0.22,8.75).

The core power subsystem number of failure modes assumption will be generated based upon STS FMEA/CIL data for the orbiter power system. As with the core avionics subsystem, the power system is considered to be similar to that of the STS vehicle with one major difference. This major difference is due to the inclusion of the fuel cells on the orbiter, which were required in order to supply power to the vehicle on orbit and during the crew return to landing. For the test problem the standard power subsystems will be assumed to be battery driven due to the expendable nature of the vehicle. Typically, fuel cells are only required for extended duration missions or on reusable vehicles [92].

In reference [145] an assessment of the orbiter power system FMEA/CIL is presented. This assessment was carried out using the same ground rules and assumptions as the reports discussed above for the avionics subsystems. The power subsystem report identifies 489 total failure modes and 163 critical items for the orbiter [145]. All of the identified failure modes are divided into six levels of criticality with loss of mission or vehicle as the top most level. A total of 12 failure modes were categorized in the top level for the orbiter power subsystem. Although the orbiter power subsystem contains fuel cells and the core stage does not, the 12 failure modes identified in the FMEA/CIL assessment will be used as the assumption for the core power subsystem. Exclusion of fuel cells means the core power subsystem is expected to have 12 modes or less based upon the orbiter assessment. Therefore, the number of modes will not be adjusted in this case in order to ensure a conservative assumption.

The number of failure modes for the standard upper stage power subsystem can now be estimated using a generic parts count approach. First, a typical list of equipment for an upper stage power subsystem must be generated. Since the power subsystem is assumed to run off of battery power instead of fuel cells, the first components on the list are batteries. Depending upon the electrical demands of the other subsystems, multiple batteries may be required. For example, the S-IVB upper stage on the Saturn V vehicle contained three 28-volt DC batteries and one 56-volt DC battery [12]. In addition to one or more batteries, a typical power subsystem equipment list will include a power control unit, signal conditioning unit, and distribution unit [92]. The generic upper stage power subsystem for the test problem will therefore be assumed to contain 6 primary components. These components include three batteries, power control, conditioning, and distribution.

The final power subsystem to consider is the integrated vehicle fluids (IVF) option for the upper stage. This option was included in order to capture a new power system concept that is currently under development. The number of failure modes assumption for the IVF option will be developed based upon a generic description of the system.

The IVF system aims to combine multiple functions on an upper stage vehicle including tank pressurization, attitude control, and electrical requirements [178]. The system utilizes the propellants already on board the stage to power an internal combustion engine, which in turn drives other elements [178]. Combining all of these functions into a single system allows for a significant reduction in vehicle dry mass, which results in an increased payload performance [177].

In reference [177], Zegler gives a description of the IVF components and a schematic illustrating the system layout. Using the provided descriptions and layout the list of primary components for an IVF system was generated. These components include gaseous oxygen and gaseous hydrogen tanks, fluid controls, hydrogen pump, oxygen pump, internal combustion engine, starter battery, attitude control unit, and instrumentation. In all, 10 primary components were identified for the IVF system. Therefore, the assumed number of failure modes for this option in the matrix of alternatives was set to ten.

5.2.5 Structure, Tanks, Other

The final subsystems to consider for the test problem include structures, attitude control, tank pressurization, etc. These subsystems are not listed directly in the matrix of alternatives and will be grouped together when considering their reliability growth. Although options are not listed in the matrix, reliability growth curves for these subsystems are needed for a more complete analysis. An extra “structure, tanks, and other” subsystem will therefore be added to the upper stage, core, and liquid boosters to capture these additional components.

The first "structure, tanks, and other" subsystem will be developed for the upper stage. As discussed previously, it is assumed that the upper stage contains liquid oxygen and liquid hydrogen propellant. The upper stage will also require its own attitude control system, which will operate after core separation. The list of additional upper stage components therefore includes the oxygen tank, hydrogen tank, propellant feed system, tank pressurization, separation system, and attitude control. This generic list leads to an assumption of 6 failure modes for the structure, tanks, and other subsystem on the upper stage.

For the liquid booster, two separate "structure, tanks, and other" subsystems will be required. These subsystems correspond to the advanced booster and the fly-back liquid booster options. For the standard advanced booster the following additional components were assumed: oxidizer tank, fuel tank, core attach struts, launch pad tie down, propellant feed system, tank pressurization, and separation. This list results in a number of failure modes assumption of 7 for the standard liquid booster. The fly-back booster option will have additional components to that of the standard liquid booster. In this case, two wings as well as control surfaces will be added to the list of components. Therefore, 10 failure modes will be assumed for the "structure, tanks, and other" subsystem in the fly-back liquid booster.

The final "structure, tanks, and other" subsystem will be contained in the core stage. As mentioned previously, the core is assumed to house liquid oxygen and liquid hydrogen propellants. The generic list of additional components therefore contains the oxygen tank, hydrogen tank, propellant feed system, tank pressurization, and attitude control. In addition, the core stage will have a load bearing intertank section, which is the primary attach point for the two boosters. From this list, a total of 7 failure modes will be assumed for the core stage "structure, tanks, and other" subsystem.

For all of the "structure, tanks, and other" subsystems the generic probability of occurrence distribution for a complex system will be used. This distribution, Beta(0.22,8.75), was used successfully during Experiment 1 and 2. In addition, it was used during the example problem in Section 4.2, which showed favorable results in terms of prediction accuracy.

5.2.6 Reliability Growth Assumptions Summary

The previous sections developed reliability growth assumptions for 20 different subsystems from the test problem matrix of alternatives. The growth assumptions addressed within these sections were the number of failure modes and probability of occurrence. These assumptions were developed using previous data, schematics, and generic system descriptions depending upon the type and heritage of each specific subsystem. In Table 21 a summary of all the reliability growth model assumptions is presented. Note that the final growth model assumption, fix effectiveness factor, will be modeled using the same uniform distribution for all the subsystems. This distribution is U(0.90,0.99) and is not shown in the table.

Two additional assumptions will be used in order to carry out the test problem. First, the total number of flights that will be projected for each vehicle architecture needs to be set. Based upon the experience gained from the previous experiments and the example problem, most vehicle growth curves tend to reach a mature level somewhere between 150 and 250 flights. In the example problem 300 total flights were used, however, the final 50-100 flights were relatively uninteresting. The number of flights for the test problem was set to 250, which is expected to capture the relevant sections of the growth curves for all vehicles in the matrix of alternatives. In order to keep the runtime at a reasonable level, the reliability growth curves will be evaluated at a total of 50 steps in time or once every 5 equivalent flights.

The final assumption for the test problem is the number of trials per step in time. This assumption represents the number of random draws that will be taken from the subsystem reliability growth curves at each step in order to generate the system level reliability distribution. During the example problem, 1000 repetitions were used successfully. However, this problem contained only a single vehicle architecture. In all, the test problem contains just over 20,000 vehicle architectures. Therefore, the number of repetitions was reduced to 750 per step in order to keep the runtime at a reasonable length.

Note that the number of repetitions may have some effect on the output mean reliability at each step. Appendix D presents a trade study exploring the effects of increasing or reducing the number of trials per step. This study ultimately helps define the optimal number of trials that will produce the correct mean value while keeping runtime to a minimum.

Table 21: Summary of the reliability growth assumptions for the test problem

Subsystem	# of Modes	Probability of Occurrence
RL-10C1	7	Beta(0.078,33.9)
RL-10C2	10	Beta(0.078,33.9)
J-2X	7	Beta(0.24,54)
US Staged Combustion Engine	10	Beta(0.026,9.85)
US Avionics	10	Beta(0.026,9.85)
US Standard Power	6	Beta(0.0421,10.04)
US Integrated Vehicle Fluids	10	Beta(0.026,9.85)
US Structure, tanks, other	6	Beta(0.0421,10.04)
STS SRB	12	Beta(0.004,11.4)
Advanced Solid	12	Beta(0.022,10.13)
LRB Gas Generator Engine	8	Beta(0.031,9.95)
LRB Staged Combustion Engine	10	Beta(0.026,9.85)
LRB Expander Engine	7	Beta(0.046,12.03)
Standard LRB other	7	Beta(0.046,12.03)
Fly-back LRB other	10	Beta(0.026,9.85)
RS-25	10	Beta(0.09,54.5)
RS-68	8	Beta(0.0759,36.02)
Core Avionics	15	Beta(0.02,10)
Core Power	12	Beta(0.022,10.13)
Core Structure, tanks, other	7	Beta(0.046,12.03)

5.3 Results

After completion of the test problem assumptions the growth models for each subsystem were setup using the Python coding language. The component and matrix row classes shown in Appendix A were then used to generate the appropriate FTA equations for every architecture. As noted in the introduction to the test problem, the three derived requirements for research objective completion must be examined. Prior to examining the results and demonstrating the utility of the CONTRAST method, these requirements will be addressed. Section 5.3.1 will examine the derived requirements and check that the method has indeed met them. After discussion of the requirements, Section 5.3.2 will discuss the results of the test problem in more detail. This section will demonstrate the utility of the CONTRAST method and will include examples of potential uses of the output reliability growth data.

5.3.1 Derived Requirements Check

The first requirement to be addressed also happens to be the easiest to evaluate; the total required runtime of the CONTRAST method. Recall that the original benchmark for the runtime was set to 64 hours, which represents the amount of time between close of business on a Friday evening and open of business on Monday morning. This benchmark was selected because during this period of time, computers are typically standing idle. For the test problem matrix of alternatives, the total runtime for all architectures was tracked. This runtime totaled only 39 hours for 20,160 architectures, which is significantly below the original benchmark. It is estimated that an additional 10,000 to 15,000 architectures could be completed within the original benchmark time frame.

The introduction of the test problem also discussed a secondary time requirement in the case where the CONTRAST method could not evaluate all of the architectures within the benchmark time frame. This secondary time frame was set to an entire week of runtime on a single computer. Using the measured runtime from the test problem, it is estimated that a single computer could produce reliability growth curves for approximately 87,000 architectures in a week's worth of runtime. Therefore, the run time of the test problem shows that the method is able to handle very large architecture spaces within a reasonable time frame. This is especially true if multiple computers can be utilized for running the reliability growth projections. Running multiple computers over a weekend or for an entire week would enable the analysis of hundreds of thousands of architectures.

After confirming that the runtime of the CONTRAST method lies in an acceptable range, the first derived requirement can be addressed. This requirement states that the method must be able to produce quantitative reliability estimates for all vehicles within the defined architecture space. For the test problem, the architecture space was defined using both heritage and new technologies. The new technologies were included in the matrix of alternatives in order to demonstrate the ability of the method to produce reliability estimates for novel concepts.

The addition of such concepts also helps to show that the third derived requirement has been met. This requirement states that the method must be flexible enough to produce reliability estimates for any vehicle concepts within the defined architecture space. Figure 48 below shows the mean reliability projection for all 20,160 architectures in the test problem matrix of alternatives. This figure ultimately confirms that the first and third derived requirements have indeed been met.

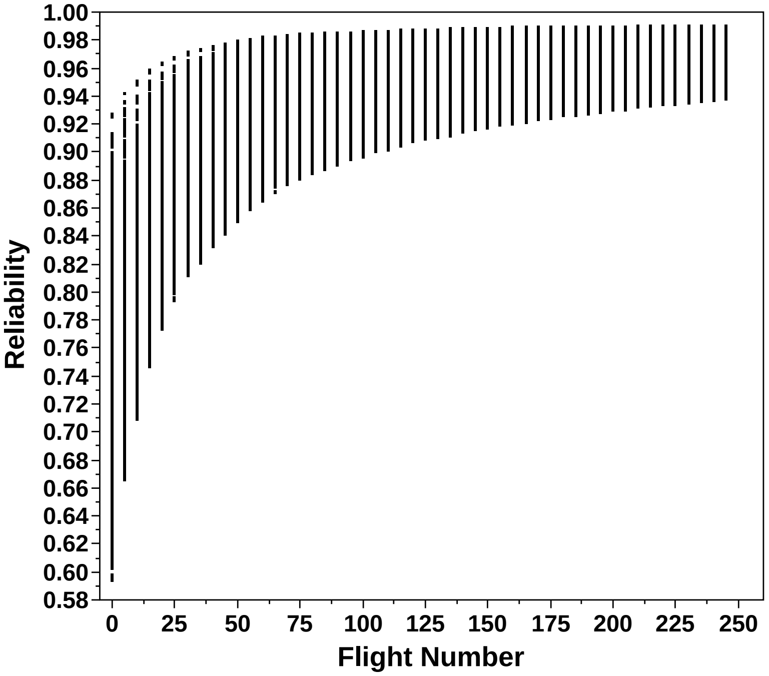


Figure 48: Reliability growth projections for all test problem architectures

The final requirement to address before examining the results in detail is the ability of the method to differentiate between unique but similar vehicles. This was the second derived requirement from Section 2.4. In order to demonstrate that this requirement has been met, multiple unique but similar vehicles will be compared using the reliability growth output.

First, consider a vehicle with a fixed booster and upper stage. For the following example, the boosters will be fixed as the STS SRB. The upper stage has been set to a two J-2X configuration with no avionics or power redundancy and a standard battery operated power system. Figure 49 and Figure 50 show the remaining architectures after the previous settings were fixed.

The first unique but similar option to consider is the trade between core engine out and no engine out. This trade is illustrated by Figure 49, which has marked the cases by core engine out. As can be seen in this figure, there is a very noticeable split in mean reliability between the engine out and no engine out cases. The lower band of cases represents no engine out capability, while the higher band represents vehicles with engine out capability. The output of the CONTRAST method has obviously captured the expected result in this case.

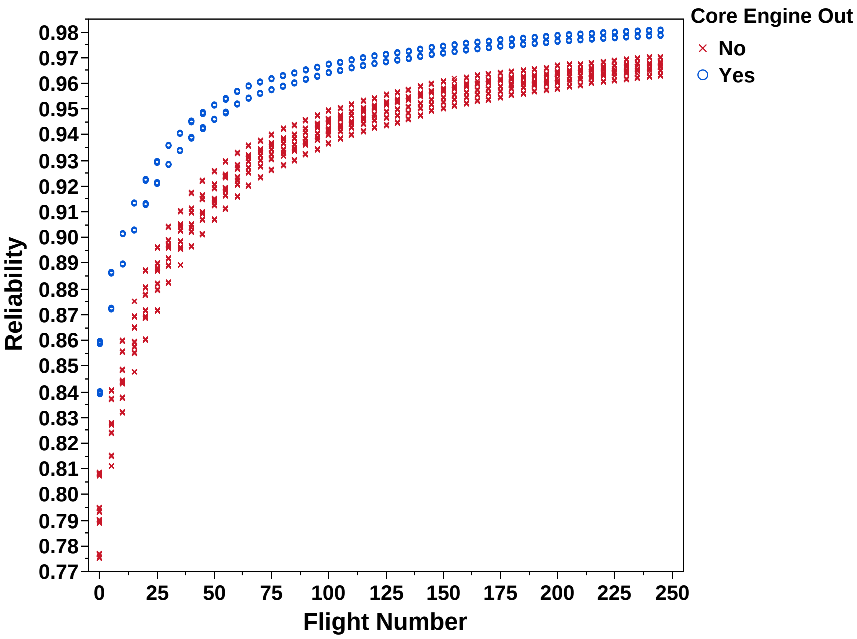


Figure 49: Reliability growth projections marked by core engine out

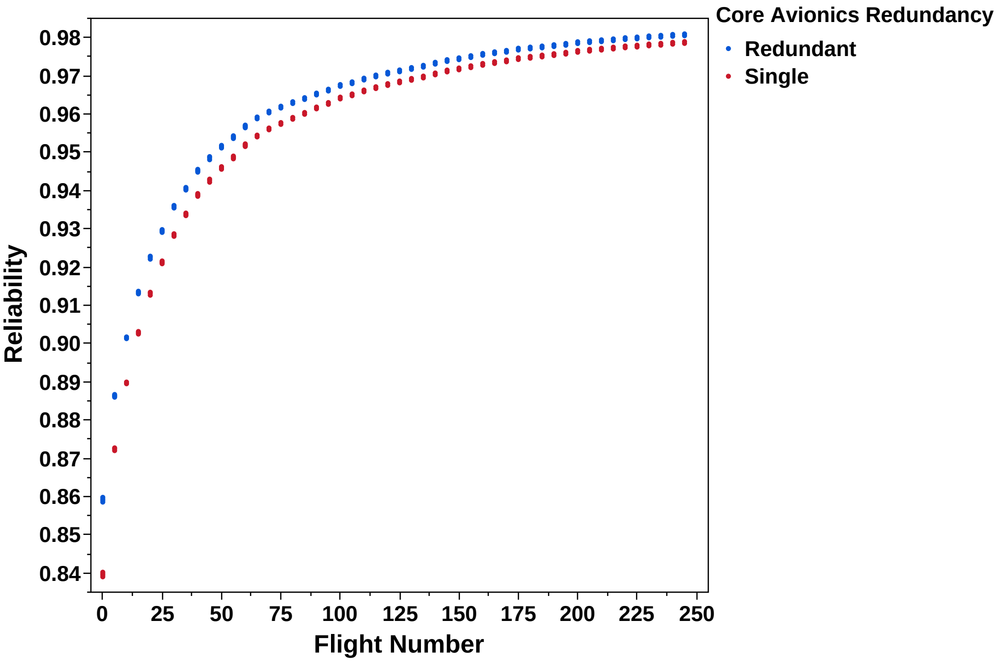


Figure 50: Reliability growth projections colored by core avionics redundancy

The second figure, Figure 50, splits out the cases that have core engine out capability. This comparison is looking at vehicles with a fixed booster and fixed upper stage, which also have a core engine out capability. The unique but similar option in this case is the core avionics redundancy, which have been colored black or gray for single and redundant, respectively. For this case there is less of a difference than what was seen for the engine out trade. However, the single and redundant options can still be easily differentiated from one another.

Two more interesting unique but similar cases can be generated by fixing the core and booster of the vehicle and allowing some of the upper stage parameters to change. These cases will look at the number of engines on the upper stage as well as the avionics redundancy. First, the upper stage engine type will be set to an RL-10C1, the power type to battery, and no redundancy will be used. The booster will be fixed to STS SRB and the core will be set to a four RS-25 engine architecture with no avionics or power redundancy.

Figure 51 shows the remaining vehicle architectures for the first case, colored by the upper stage number of engines. In this figure a very clear trend can be seen, which shows an incremental decrease in expected mean reliability with an increase in number of engines. This is an intuitive result as an increase in number of engines increases the number of points of failure for the stage.

The next upper stage unique but similar case can be seen in Figure 52. This figure illustrates the difference between upper stages with and without avionics redundancy. In this case the number of engines has been set to four. As seen in the figure, this case is not nearly as pronounced as the number of engines case. However, the output still shows a discernable difference between the two options.

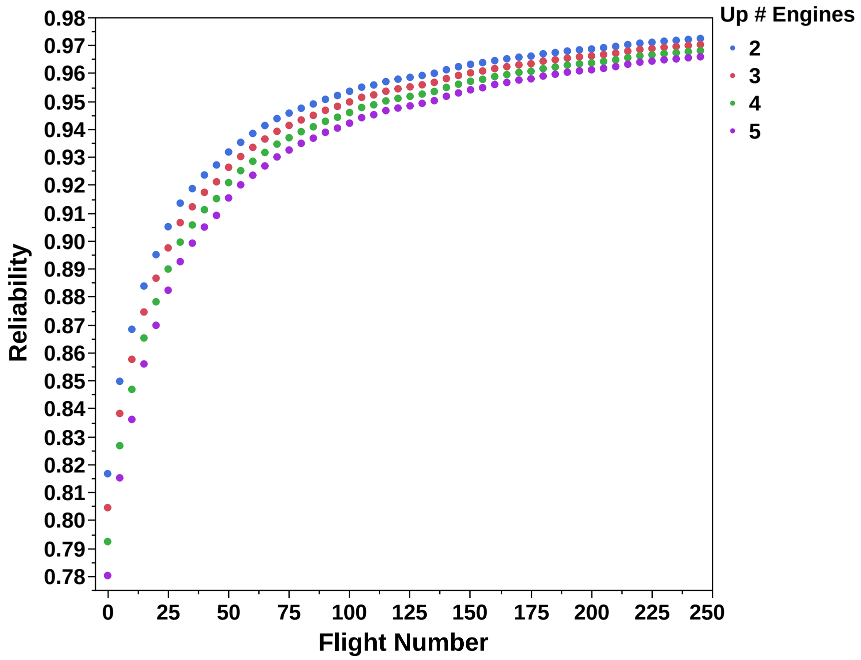


Figure 51: Reliability growth projections colored by upper stage number of engines

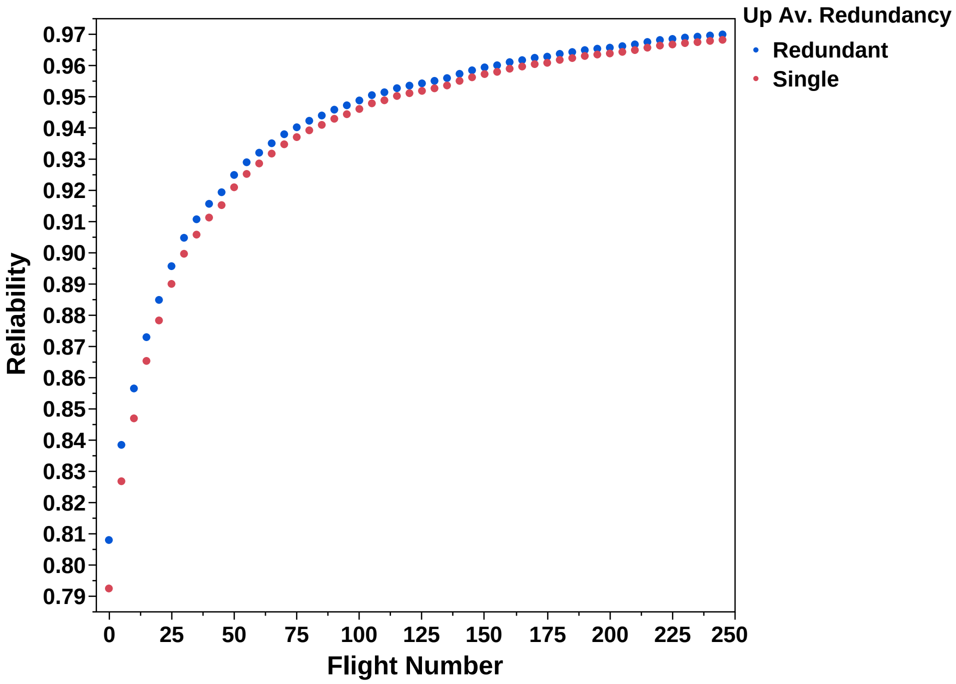


Figure 52: Reliability growth projections colored by upper stage avionics redundancy

The four examples given above are just a few of many unique but similar concepts that can be compared using the output of the CONTRAST method. The results of the test problem can therefore be used to conclude that the original derived requirements to meet the research objective have been met. The test problem results have given quantitative estimates for all vehicles in the architecture space, including a few new concepts. These results have also demonstrated the CONTRAST method's ability to compare unique but similar vehicle concepts.

5.3.2 Detailed Test Problem Results

After addressing the derived requirements for objective completion, the test problem results can be looked at in further detail. This section will first explore the data that was generated for the architectures within the test problem matrix of alternatives. Through data exploration, preferred analysis views for the CONTRAST method's results will be identified. Following the simple data exploration a discussion on more complex uses of the output data will be presented. This discussion will include the evaluation of the probability of meeting a desired reliability requirement in a set number of launches or the required time to reach said requirement. It will also present an application of the method in which block upgrades can be analyzed.

5.3.2.1 Data Exploration

Exploration of the test problem data will start from Figure 48, which plots the mean reliability versus flight number for all of the architectures. From this plot, simple data filters can be used to view how the full design space is partitioned based upon the various architecture options. The initial filtering operations will allow the analyst to quickly view the effects of each architecture option on reliability. The first filtering example will look at the liquid versus solid booster types.

In Figures 53 through 56, four plots are given of all of the reliability growth data with different booster selections. The first two, Figure 53 and Figure 54, show all of the architectures with solid boosters. The other two, Figure 55 and Figure 56 show all of the vehicles with liquid boosters. When moving from the STS SRB to the advanced booster, the section of selected points shifts slightly downward. This immediately tells the analyst that the highest reliability vehicles within the data utilize STS derived solid boosters. On the other hand, the two liquid booster plots do not show a recognizable difference between the advanced liquid booster and the fly-back liquid booster. It does however show that the liquid booster cases will tend to have lower reliabilities than the solid cases.

The basic filtering results shown in Figures 53 through 56 are very intuitive. Due to its flight heritage, the STS SRB option is expected to perform best in terms of reliability, while the advanced solid falls slightly below. Both of the liquid boosters require development of a new liquid rocket engine, which explains the generally lower reliability values. Note that some overlap between the four options does exist. Although the STS SRB vehicles tend to be the highest reliability, there are many cases in which an advanced booster or liquid booster architecture will perform better in terms of reliability.

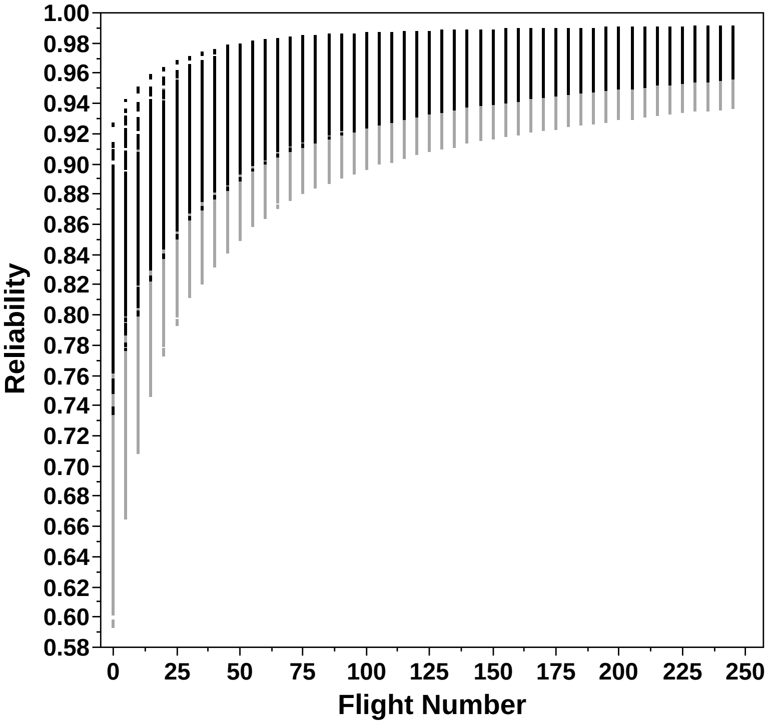


Figure 53: Reliability growth projections with STS SRB highlighted

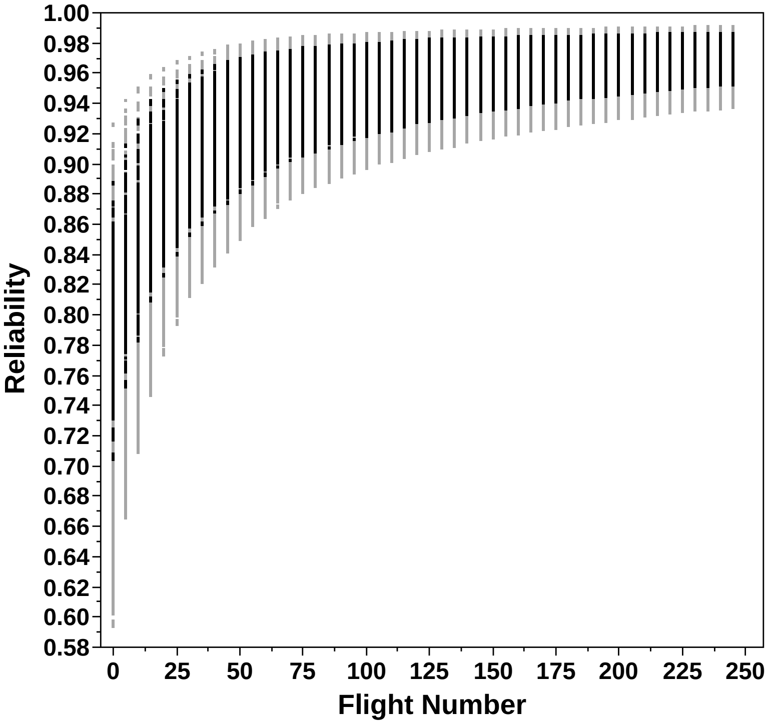


Figure 54: Reliability growth projections with Adv. SRB highlighted

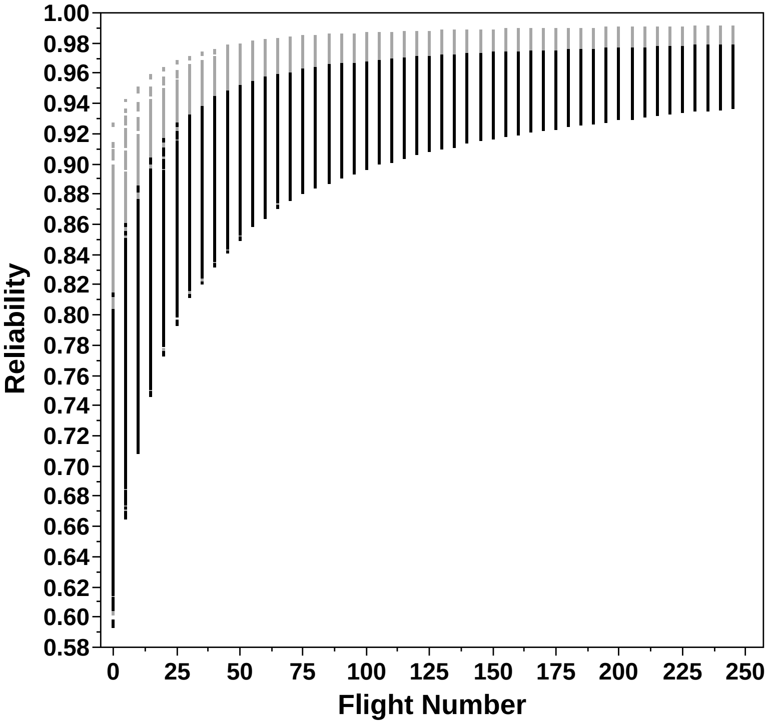


Figure 55: Reliability growth projections with LRB highlighted

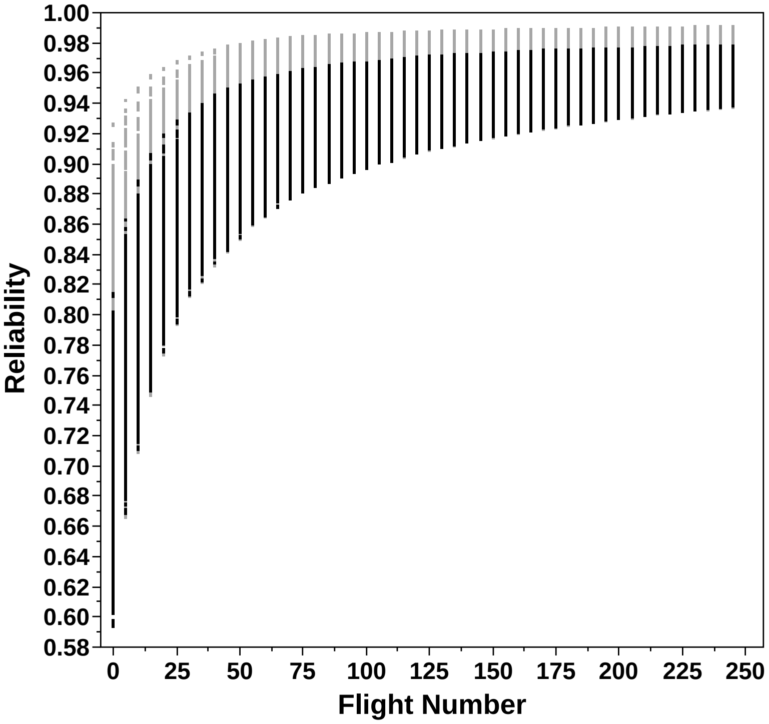


Figure 56: Reliability growth projections with fly-back LRB highlighted

An example from the initial data filtering that shows very little difference is the core avionics and power redundancy. In this case, both the single and redundant options spread across the entire range of reliabilities. This case is a prime example of an option that may be dismissed for the time being because it seems to have little effect on the overall vehicle reliability.

Figure 57 shows the selection of all vehicles with single core avionics and single core power subsystems. The second plot, Figure 58, shows the selection of all the vehicles with redundant avionics and redundant power systems. These figures show only a very small difference at the very top and bottom of the reliability growth curves. The first figure does not include the vehicles with the highest reliability, while the second does not include the vehicles with the lowest reliability. Since each of the selections show a wide spread in reliability value, this option can be considered less vital to achieving high reliability.

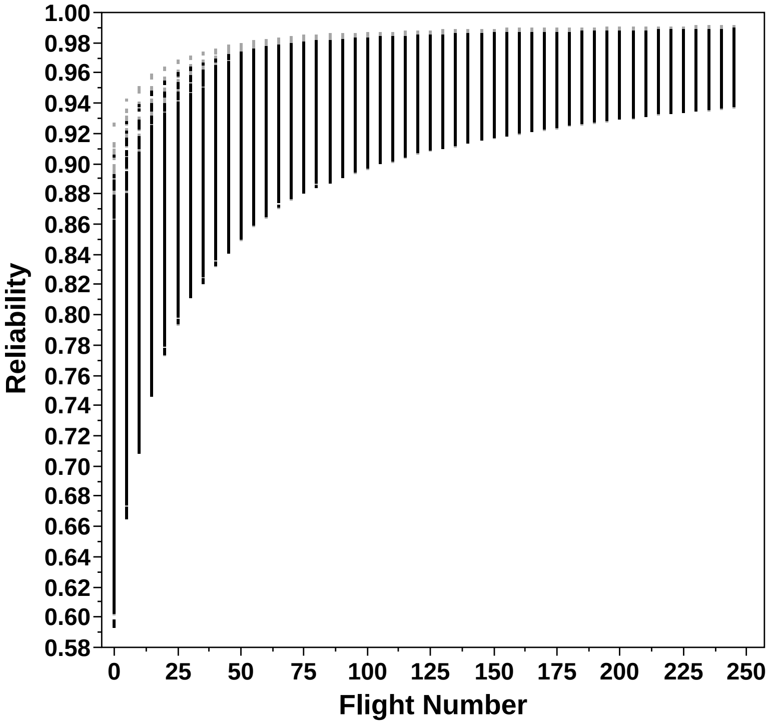


Figure 57: Reliability growth projections with no redundancy

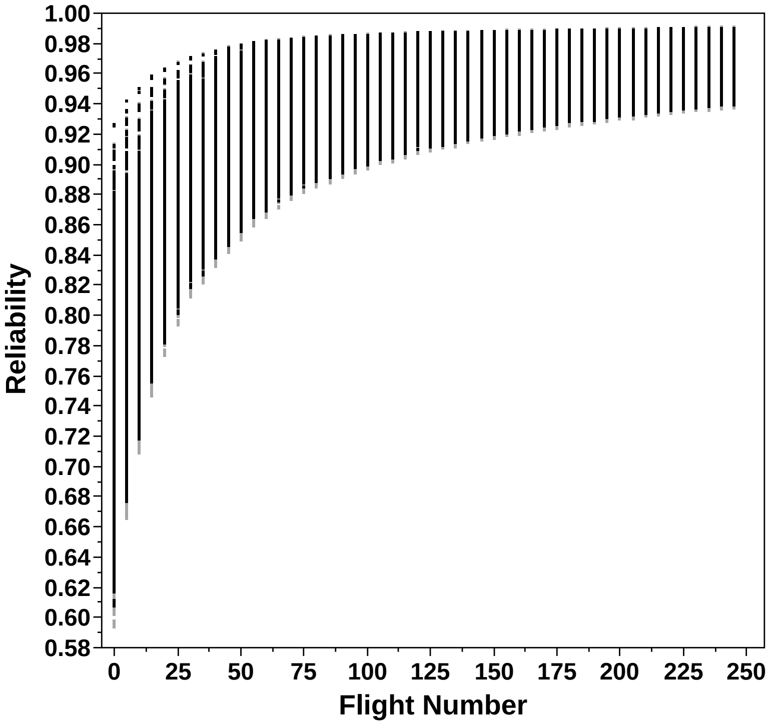


Figure 58: Reliability growth projections with full redundancy

Similar to the core avionics and power redundancy, the upper stage avionics and power redundancy did not show a significant trend when using the simple filters. In addition, the upper stage power type showed only a slight difference with all of the highest reliability architectures containing a standard power system. The upper stage number of engines did reveal a noticeable trend as the higher number of engines tended towards the low end of the reliability range. This trend was shown previously in Figure 52.

Through additional filtering, two of the core stage options were shown to have a fairly large impact upon the vehicle reliability. The first option, core number of engines, showed a narrow band of cases through the middle of the reliability range for a four engine architecture. The five engine architectures actually spanned the entire reliability range with cases at the very highest reliability and cases at the lowest. This trend ultimately illustrates the importance of the second core stage option, engine-out capability. Due to the fact that engine-out was only enabled for 5 engine core stages, the reliability of these stages depended heavily on the engine-out option. In the cases where no engine-out was selected, the 5 engine architectures tended to be less reliable than the 4 engine architectures. On the other hand, if engine-out was selected, the 5 engine core stage architectures were able to reach the highest reliability values.

Preliminary filtering of the reliability growth data can help to identify options within the matrix of alternatives that may have the largest effect on vehicle reliability. This filtering approach can also serve as a verification step prior to detailed data analysis. Expected trends within the data can be checked in order to verify that the analysis was completed correctly. If a counter intuitive trend is identified, the analyst can focus attention on this trend and determine whether a batch of cases needs to be re-run through the analysis.

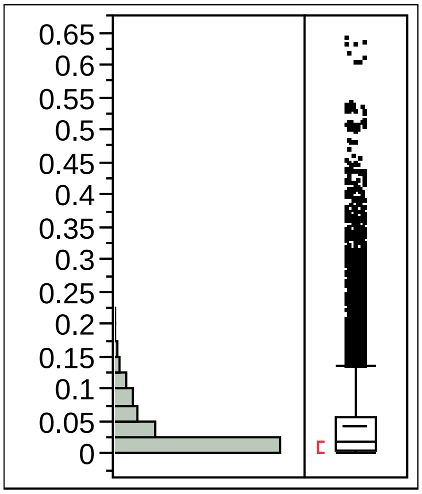
5.3.2.2 Probability of Meeting a Specific Requirement

The previous section presented a very simple first step for exploring the output data from the CONTRAST method. After initial exploration of the data, there are multiple more detailed analyses that will be very useful to the reliability analyst. The first of these analyses is to determine the probability of a vehicle meeting a specific reliability requirement. The results from this analysis will show which architectures can meet the reliability requirement that has been laid out for the vehicle. In addition it will allow the analyst to identify architectures that have the highest probability of meeting or surpassing the requirement.

For the test problem results a reliability requirement will be assumed in order to demonstrate the utility of this approach. A probability of failure of 1 in 100 flights will be used as the reliability requirement. This number stems from the estimated final probability of LOC for the Space Shuttle [78].

In order to identify the vehicles that reached the reliability requirement, the maximum values for all steps in time for all architectures were filtered. This filter resulted in the identification of 19,806 out of 20,160 architectures that reached a maximum expected reliability of 0.99 or higher at some point during the flight history. These architectures were then separated from the primary set of data for further analysis.

First, the expected reliability distributions of the architectures were examined at the last equivalent flight. This analysis identifies the probability of reaching the reliability requirement at maturity for each vehicle. Figure 59 shows a distribution of the probability of meeting the reliability requirement at flight 250 for all 19,806 architectures. As seen in the figure, 75% of these architectures have less than a 5% probability of meeting the reliability constraint at the last flight. Within the defined architecture space, the maximum probability of meeting the 0.99 reliability requirement at the last equivalent flight is 64%.



Quantiles		
100.0%	maximum	0.64078
99.5%		0.40757
97.5%		0.21521
90.0%		0.11166
75.0%	quartile	0.05689
50.0%	median	0.01826
25.0%	quartile	0.00512
10.0%		0.00132
2.5%		0.00014
0.5%		4.87e-6
0.0%	minimum	9e-11

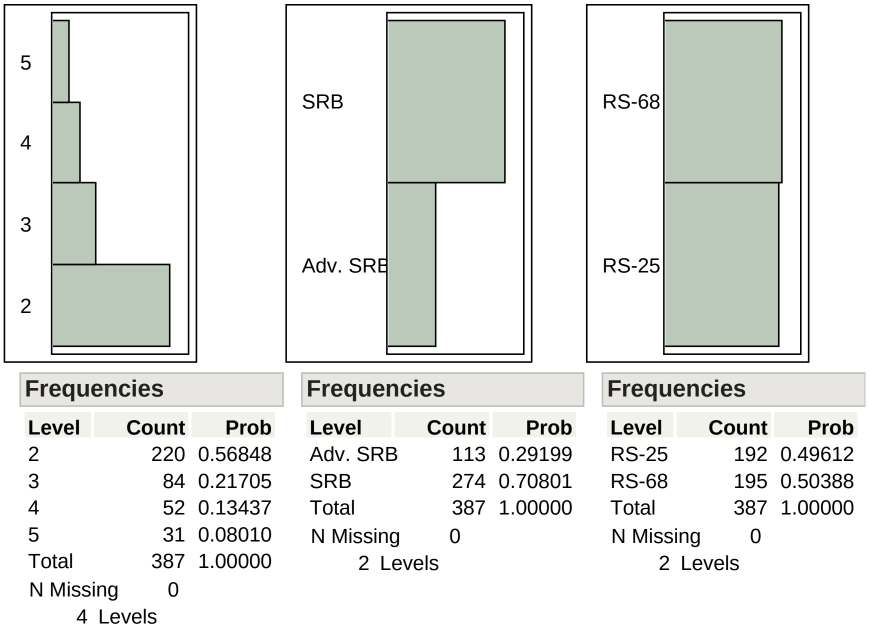
Figure 59: Probability of meeting a specific reliability requirement at maturity

To reduce the number of architectures being considered another constraint will be applied that states that the vehicles must have at least a 25% probability of meeting the reliability requirement at maturity. After enforcement, 387 architectures were found that met this constraint. From these architectures, a few of the options from the matrix of alternatives can be highlighted as important for achieving the reliability requirement.

First, the core engine options suggest that the core engine out option is key for achieving high reliability. Every vehicle within the remaining architectures utilized a 5 engine core with engine out capability. It is interesting to note, however, that the engine type did not seem to make a difference as there were nearly identical numbers of RS-25 and RS-68 engines.

Another option that fell out after enforcing the 25% probability of achieving the reliability requirement at maturity was the booster type. None of the 387 architectures utilized a liquid booster, while a majority of the architectures used the STS derived SRB. A quick test revealed that 90% of the liquid booster cases provided less than a 5% chance of meeting the 0.99 reliability requirement at maturity.

Figure 60 below shows a few of the example distributions from the 387 architectures. The left most distribution illustrates that a majority of the vehicles that met the 25% constraint utilized upper stages with only 2 engines. The middle distribution shows the split of vehicles using the STS derived SRB versus the advanced SRB. Finally, the right most distribution shows the nearly even split between the RS-25 and RS-68 core stage engines.



Frequencies

Level	Count	Prob
2	220	0.56848
3	84	0.21705
4	52	0.13437
5	31	0.08010
Total	387	1.00000
N Missing	0	

4 Levels

Frequencies

Level	Count	Prob
Adv. SRB	113	0.29199
SRB	274	0.70801
Total	387	1.00000
N Missing	0	

2 Levels

Frequencies

Level	Count	Prob
RS-25	192	0.49612
RS-68	195	0.50388
Total	387	1.00000
N Missing	0	

2 Levels

Figure 60: Example distributions for 387 vehicle architectures

Next, let us consider a few specific vehicle architectures from the 387 that have been filtered. These architectures will illustrate comparisons that can be made between specific vehicles. Table 22 lists three different vehicle architectures along with their specific vehicle options. The three architectures in the table represent vehicles with differing upper stage engines, upper stage power system types, solid boosters, core stage engines, and redundancy options. Note that these vehicles were chosen because they represent different unique vehicle architectures, but they all show nearly the same probability of meeting the reliability requirement at maturity. This was done in order to illustrate an important comparison between the CONTRAST method and the existing methods discussed in Section 2.3.

First the mature reliability distributions for each of these vehicles will be considered. This will give an estimate for the probability that each vehicle will meet the defined reliability requirement of 0.99. Figure 61 below shows the three vehicle reverse cumulative distribution functions at the last flight. As can be seen in the figure the CDFs for these vehicles are similar, with only slight deviation between 0.95 and 0.98. The assumed reliability requirement line is plotted in the figure, which shows that all three of the vehicles have around a 30% chance of meeting or exceeding this requirement. These exact probabilities are 29.81%, 31.96%, and 31.05% for vehicle 1, 2, and 3, respectively.

Table 22: Three specific vehicle architectures for comparison

Option	**Vehicle 1**	**Vehicle 2**	**Vehicle 3**
Architecture #	9	9790	18501
Upper stage engines	2 RL-10C1	2 RL-10C2	2 New Staged Comb.
Upper stage power	Single	Redundant	Single
Upper stage avionics	Single	Redundant	Single
Upper stage power type	Battery	Battery	IVF
Booster type	SRB	Adv. SRB	SRB
Core engines	5 RS-25	5 RS-68	5 RS-68
Core engine out	Yes	Yes	Yes
Core power	Redundant	Single	Redundant
Core avionics	Single	Redundant	Single

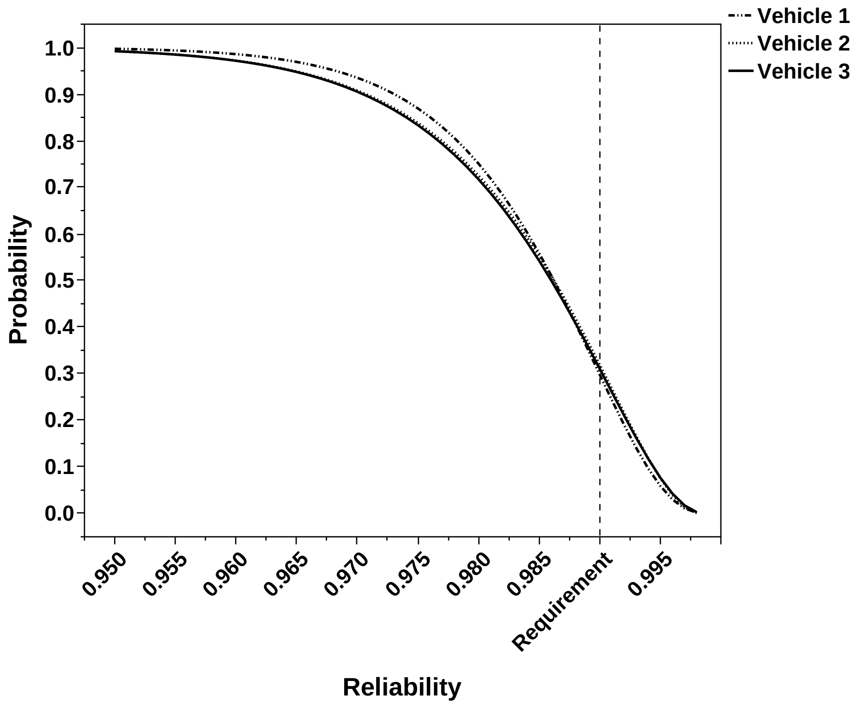


Figure 61: Reverse cumulative distribution functions for 3 vehicles at flight 250

As mentioned above, these vehicles were selected because they reach nearly the same probabilities of requirement achievement at maturity. This case can therefore be compared back to the state-of-the-art reliability tool, FIRST, which was discussed in Section 2.3. In Section 2.3 the limitations of the mature estimates from the FIRST tool were discussed. One of the primary issues with mature estimates will be illustrated using the three vehicles in Table 22.

Figure 62 below shows the data for these vehicles in the form of box plots at flight 250. This alternative plot shows the mature reliability of the vehicles in the same form as the FIRST tool. Using these mature reliability distributions offers very little information to differentiate between the three concepts. As can be seen in the plot, the variability of vehicle 1 seems to be slightly lower than the other two vehicles. However, vehicle 1 has the lowest probability of reaching the reliability requirement. The results in the mature reliability form used by FIRST would therefore suggest that any of the vehicles would be a satisfactory selection.

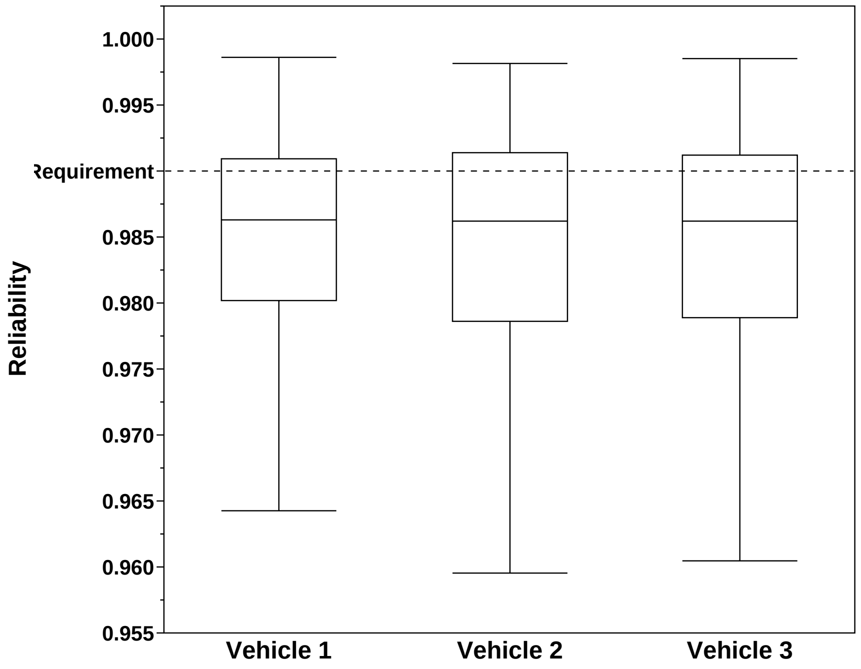


Figure 62: Reliability box plots for three vehicles

Next, the rest of the reliability growth results produced by the CONTRAST method will be considered. These results will give additional information about each concept that would not be available when using existing tools. To illustrate the additional results, the baseline 25% probability of meeting the reliability requirement will be examined. In this example, let us assume that a vehicle has reached its "mature" state when it has at least a 25% chance of having a reliability that is greater than or equal to the defined requirement. For each of the vehicles the reliability growth results can then be used to determine the number of flights that are required to ensure this 25% probability of meeting the 0.99 requirement. Table 23 gives the number of flights required for each vehicle to reach the desired probability.

Table 23: Flights to reach 25% probability of meeting the 0.99 requirement

Vehicle Number	**Flights to 25% Prob. of Success**
1	205
2	190
3	165

Now that the reliability growth data has been taken into account the three vehicles are easier to differentiate. As seen in the table the vehicles have fairly significant differences in their number of flights required to reach the 25% probability of success benchmark. The first vehicle has the longest required time at 205 flights, which is 40 flights higher than vehicle 3. Note that vehicle 1 may have been the most desirable choice when only considering the mature reliability because of its lower variability. However, if this vehicle had been chosen it would have required a much higher number of equivalent flights to reach the desired reliability level. The reliability growth output therefore suggests that vehicle 3 is the most desirable option because it will mature at a much faster rate. The ability to identify the amount of time or equivalent flights required to reach the reliability requirement was one of the primary motivators for this research. The three vehicles above are just one example of the additional capabilities that the CONTRAST method provides beyond that of current state-of-the-art tools.

5.3.2.3 Required Flights to Meet a Specific Requirement

The previous section presented an example using the reliability growth output to identify the number of flights required to reach a specific level of reliability. This section will continue to look at the required number of flights but from a different angle. The previous example was looking at the number of flights required to reach maturity. The example presented below will look at the number of flights required to reach a go/no-go type reliability requirement for a first operational flight. This estimate will give the analyst an idea of the time required in development and testing to deliver the vehicle to its first flight.

First, the reliability requirement for first flight needs to be defined. This requirement can be determined based upon the estimated reliabilities of historical vehicles at their first operational flight. For the Saturn V launch vehicle only 2 unmanned test flights were conducted prior to the first manned launch [61]. At the time of the first manned launch the reliability of the vehicle was predicted to be as low as 0.75%, meaning a probability of LOV of 1 in 4 flights [63].

The most recent U.S. manned launch vehicle, the STS, faired a little better than the Saturn V in terms of predicted reliability at first flight. No unmanned test flights were performed prior to the first manned flight. The reliability of the vehicle for this flight was estimated to be approximately 90% [78]. This represents a probability of LOV of 1 in 10 flights. For the example in this section, the desired reliability at first flight will therefore be set at 0.9. Since this is a relatively low value it will be assumed that a vehicle must have at least a 95% chance of meeting or exceeding the reliability requirement in order to qualify for its first flight.

The first view to look at in this case is the entire grouping of architectures similar to Figure 48. As a preliminary exploration, the y-axis can be set with a minimum of 0.9 and the architectures can be colored by specific options. This will give an idea of which options tend to have an effect on the required number of equivalent flights to meet the first operational flight requirement. Figure 63 below shows all of the architectures colored by core engine out. Figure 64 shows all of the architectures colored by booster type.

Figure 63 shows a very pronounced trend after coloring all the architectures by core engine out. As was shown in the previous section, all of the highest reliability architectures utilize core engine out capability. The lowest reliability vehicles with engine out still tend to be in the middle of the pack in terms of reliability. The non-engine out cases on the other hand tend to be much lower reliability. However, there is some overlap between the two options where non-engine out vehicles will surpass engine out vehicles in terms of reliability. This trend is very intuitive because engine out capability is a primary method for increasing vehicle reliability.

The second plot, Figure 64 shows all of the architectures colored by booster type. In this case another noticeable trend is revealed. The first booster type is the STS based SRB, which due to its extensive flight history is expected to be high reliability. The advanced booster cases fall in the upper middle section of the plot and share some overlap with the STS derived booster. Both of the liquid boosters lie on the bottom half of the architectures with some overlap with the solid boosters.

Since the liquid boosters tend to have lower reliability, they also require more equivalent flights to reach the 0.9 reliability level. In the bottom left hand corner of the plot all of the solid booster cases fall between 0 and 30 or 40 flights at the 0.9 level. The liquids on the other hand range between 20 and 110 equivalent flights. This trend raises an interesting point regarding the development of new technologies. The STS SRB is regarded as the highest reliability because of its flight history with the advanced booster following in close second. The liquid boosters however, represent major development programs including new liquid rocket engines. Therefore, it is expected that many more equivalent flights will be needed to meet the first flight reliability requirement.

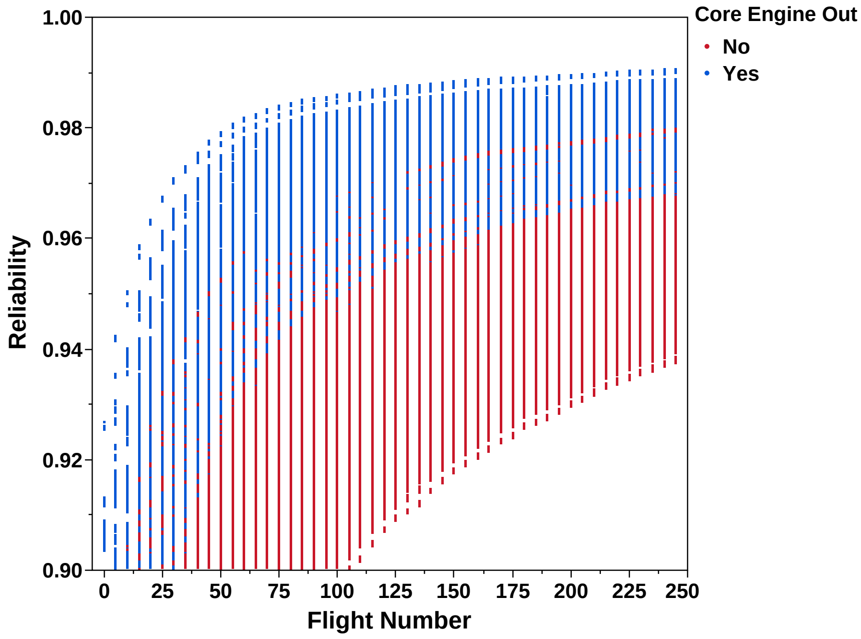


Figure 63: All vehicle architectures colored by core engine out

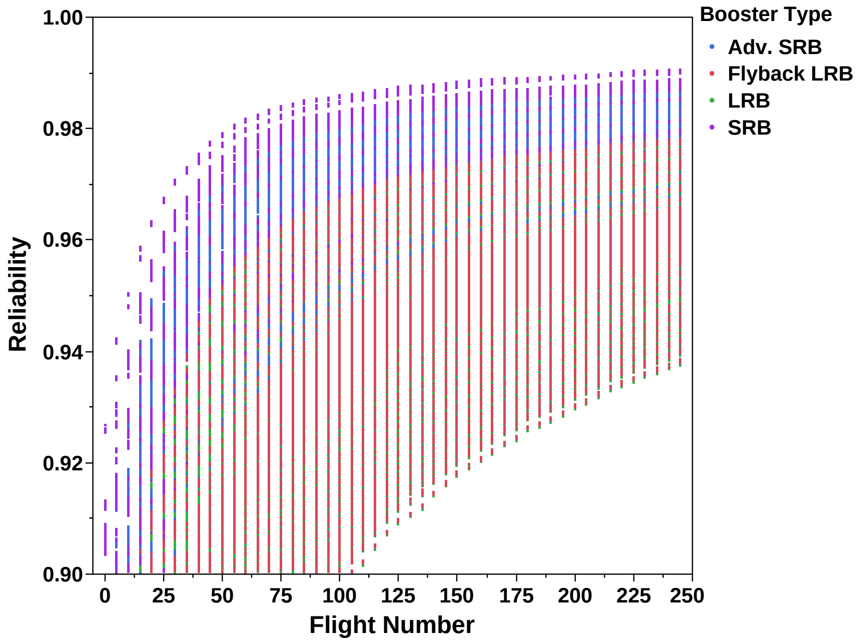


Figure 64: All vehicle architectures colored by booster type

The trends observed in the initial explorations can now be examined in more detail. As mentioned above the core engine out and booster type options showed the most prominent reliability trends. Therefore, a set of vehicles will be selected, which will represent the different architecture options. The baseline vehicle will be representative of the initial configuration of the SLS vehicle. Three other architectures will be added to capture the addition of advanced solid boosters, standard liquid boosters, and fly-back liquid boosters.

Each of these four vehicles will be analyzed with and without engine out capability. In the interest of consistency, all vehicles will be assumed to have single avionics and single power subsystems in the upper stage and core. The equivalent flight for each vehicle will be recorded when the 95% probability of meeting the 0.9 reliability requirement is met. Table 24 lists the defaulted options used for all configurations. The engine out and booster type options can be seen in Table 25, which also displays the results for all 8 configurations.

Table 24: Defaulted options for time to first flight study

Option	**Default Value**
Upper Stage Engines	4 RL-10C1
Upper Stage Avionics	Single
Upper Stage Power	Single
Upper Stage Power Type	Battery
LRB Engines	2 Gas Gen.
Core Engines	5 RS-25
Core Avionics	Single
Core Power	Single

The required number of equivalent flights to reach the 95% requirement is tabulated below for all 8 vehicle configurations. As seen in the table, the effects of core engine out and booster type on time to first flight are very clear. The first four vehicle cases have no engine out capability with each of the four booster options. The STS SRB offers the lowest number of required equivalent flights at 131, which is over 70 flights fewer than the two liquid booster options. It is interesting to note that both the liquid boosters have nearly identical results for the number of flights required. This suggests that a selection between the two liquid boosters will have little effect on the time required to reach the first operational flight.

The final four cases again utilize all of the booster options, but also include core stage engine out. In this case there is a slightly smaller difference between the STS SRB and the liquid boosters at 50 additional flights. These four cases show the large effect that core engine out has on reliability. Comparing vehicle 1 to vehicle 5, we see a difference of 83 flights to reach the first flight criterion. When considering the liquid booster cases this difference increases to nearly 100 flights. These results show that enabling a core engine out capability will help to drastically reduce the amount of time required for the vehicle to reach the first flight reliability criterion.

Table 25: Flights required to meet the 95% requirement for 8 vehicle configurations

Architecture	Core Engine Out	Booster Type	Flights Required
Vehicle 1	No	SRB	131
Vehicle 2	No	Adv. SRB	139
Vehicle 3	No	LRB	205
Vehicle 4	No	Fly-back LRB	204
Vehicle 5	Yes	SRB	58
Vehicle 6	Yes	Adv. SRB	64
Vehicle 7	Yes	LRB	109
Vehicle 8	Yes	Fly-back LRB	108

The vehicle results shown above illustrate the importance of two of the matrix options when considering the number of equivalent flights to reach the assumed first flight requirement. However, it is still of interest to explore the effects of the parameters in Table 24, which were defaulted. In order to explore the effects of all of the matrix options the full set of vehicle architectures was first filtered to include only the vehicles that were able to meet the 95% criterion. This filter resulted in 17,848 out of 20,160 that had at least a 95% chance of being at least 90% reliable at some point in their flight history. For each of these architectures the earliest flight at which the criterion was met was recorded.

A quick look at the minimum number of flights required for the 17,848 architectures revealed that only 12% of the vehicles were able to meet the requirement in fewer than 100 flights. In addition, approximately 10% of these vehicles met the requirement at flight 225 or later. The mean number of required flights for this data set was 161 equivalent flights.

Next, to explore the effects of all of the architecture options a simple neural network fit was utilized. A neural network is a type of surrogate modeling technique, which utilizes a network of activation function driven nodes to connect input to output. Using the architecture data, the node weighting values are optimized such that the output of the network closely matches the actual output value from the data table. In this case the input data consisted of the architecture options, while the output was the required number of flights. The neural network fit ultimately allows for the use of a profiler, which shows the effects of each architecture input on the number of required flights to reach the requirement.

The first profiler in Figure 65 shows the upper stage options versus the predicted number of flights to reach the 95% requirement. The five upper stage options can be seen listed across the x-axis at the bottom of the figure. The current selections for each option are listed in red. As seen in the profiler plots, the two redundancy options are very flat and thus have little effect on the number of flights. The power system type option also has little effect on the output.

The engine options on the other hand do show a noticeable trend. First, the engine type option shows a slight dip towards the RL-10C1 and RL-10C2. This shows that the RL-10 engine options may be more desirable when considering the first flight requirement. Due to the fact that the J-2X and new staged combustion engine represent new development programs, this observation is very intuitive. The upper stage number of engines option shows the largest trend with the number of flights increasing linearly with the number of engines. This observation is also intuitive as reliability is generally expected to decrease with the addition of more components in the system.

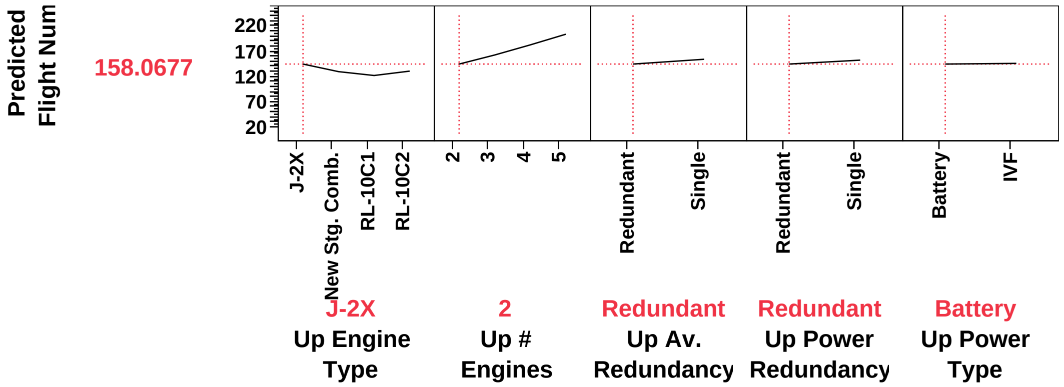


Figure 65: Prediction profiler for upper stage options

The second prediction profiler shows the booster and core options from the matrix of alternatives. The three booster options are seen at the far left of the profiler followed by the five core options. This profiler first shows that the STS SRB performs best in terms of number of flights to reach the requirement. The "None" options for liquid engine type and LRB number of engines are both the lowest points in their associated profiler plots. These points correspond to both the SRB options. The booster type profiler shows the liquid boosters being nearly equal in terms of number of required flights. The advanced solid falls slightly below the liquids and the STS SRB achieves the lowest level.

The core options are all very flat besides the engine out profiler. As shown in Table 25, the engine out selection seems to have the largest impact on the number of required flights. The second profiler confirms the previous observation, showing a very steep decline in number of flights when moving from "No" to "Yes" for the engine out selection. The remaining core options show very little effect on the number of flights. The core number of engines option shows a slight decrease between 5 and 4 engines, however if engine out was switched to "Yes" this decrease would be negligible. Both of the engine options for the core lie at the same level with a very slight advantage to the RS-68 engine. The avionics redundancy shows a slight improvement with the selection of "Redundant" but the power redundancy shows almost no effect.

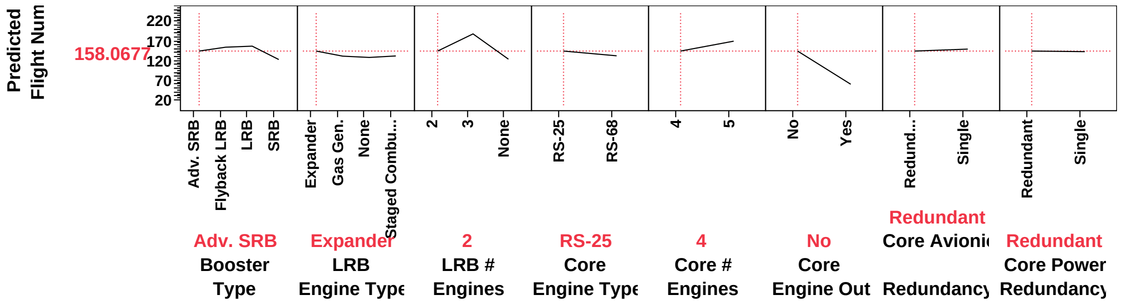


Figure 66: Prediction profiler for booster and core options

The examples above illustrate the importance of the various architecture options when considering the required time to reach a first flight reliability requirement. Consideration of the development for each of the vehicle components is therefore very important to the eventual success of the vehicle. The selected architecture options will ultimately affect the vehicle's ability to meet or exceed the reliability requirements that have been laid out for the program.

Due to the effects of technology development on reliability growth a block upgrade type approach can be taken for launch vehicles. In this approach the heritage systems, such as the STS SRB in the example, are used on early versions of the new vehicle as the other technologies are being advanced. In this case a trade is made during the early flights between performance and reliability. Using the heritage systems allows the vehicle to progress more rapidly towards first flight. Then later on in the program the vehicle is upgraded in order to increase its overall performance, hopefully at little to no cost in terms of reliability. This block upgrade approach is very significant to the SLS program and is of interest for further analysis. Due to the fact that the CONTRAST method is well suited to look at block upgrades and their effects on reliability, the next section will present analysis of the proposed SLS block upgrades.

5.3.2.4 Analysis of Block Upgrades

Development of the SLS launch vehicle is currently underway. The SLS will act as NASA's heavy lift launch vehicle, which will enable human exploration to the moon and beyond [124]. The development of the SLS will utilize a block upgrade approach, which focuses on upgrades to the upper stage and boosters. The initial configuration of the vehicle utilizes STS derived SRBs along with a Delta-IV based cryogenic upper stage [124]. Proposed upgrades to the upper stage include a four engine RL-10 configuration, or a higher thrust configuration with 2 J-2X engines [32, 151]. Other potential upgrades to the SLS include advanced solid boosters and advanced liquid boosters [10, 33]. An illustration of the proposed upgrades for the program can be seen in Figure 67.

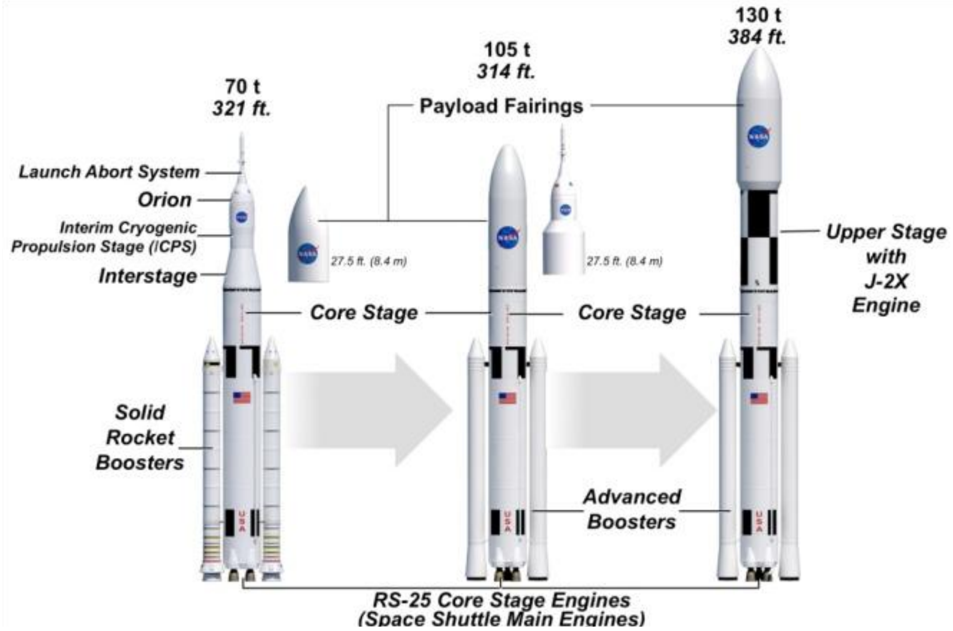


Figure 67: Notional diagram of SLS block upgrades [102]

The goal of this section will be to explore the various options for SLS block upgrades and identify any significant effects on vehicle reliability growth. In addition to exploring the different options for upper stages and boosters, the study will look at the order in which the block upgrades are implemented. As shown in the previous section, various architecture options will have large effects on the maturation rate of the vehicle. Therefore, it is expected that the order of implementation of the SLS upgrades will have an effect on the reliability growth of the vehicle. The study in this section will evaluate the order of the upgrades in order to determine the most desired path for block upgrades in terms of vehicle reliability.

To begin the study the baseline vehicle and any upgrade options must be identified. The baseline vehicle, as seen in Figure 67, utilizes STS derived SRBs and an interim cryogenic upper stage that is based on the Delta-IV [124]. The core stage of the baseline vehicle uses 4 RS-25 engines and for the purpose of this study will be assumed to have no engine out capability. The avionics and power redundancy options for core and upper stages will be set to redundant for all cases. Redundancy is assumed because the vehicle will ultimately be human rated. Table 26 lists the matrix of alternatives options that define the baseline SLS concept.

Two options will be assumed for the upper stage upgrades. The first will be a 4 RL-10 engine configuration, which will also be assumed to contain an IVF power system. This configuration is based upon the proposed dual use upper stage [32]. The second option is a J-2X driven design, which will also be assumed to contain the IVF power system. The options from the matrix of alternatives that describe these stages can be seen in Table 27.

Finally, two options for booster upgrades will be included. The first is the advanced solid booster, which was discussed in Section 5.2.3. This SRB represents an incremental improvement in performance above the STS based SRB. The second booster option will be a liquid rocket booster, which will be assumed to have 2 gas generator engines. The LRB configuration is based upon an F-1 derived liquid rocket booster that is currently under consideration [34]. The options selected in the matrix of alternatives for the two boosters can be seen in Table 28.

Table 26: Matrix of alternatives options for baseline SLS configuration

Option	**Baseline**
Upper Stage Engines	2 RL-10C1
Upper Stage Avionics	Redundant
Upper Stage Power	Redundant
Upper Stage Power Type	Battery
Core Engines	4 RS-25
Core Avionics	Redundant
Core Power	Redundant
Booster Type	SRB

Table 27: Upper stage block upgrade options for SLS

Option	**Upper Stage 1**	**Upper Stage 2**
Upper Stage Engines	4 RL-10C2	2 J-2X
Upper Stage Avionics	Redundant	Redundant
Upper Stage Power	Single	Single
Upper Stage Power Type	IVF	IVF

Table 28: Booster block upgrade options for SLS

Option	**Booster 1**	**Booster 2**
Booster Type	Advanced SRB	LRB
Booster Engine Type	N/A	Gas generator
Booster # Engines	N/A	2

After defining the upper stage and booster upgrade options the reliability growth projections for each of the vehicles can be compared. These initial comparisons will look at the growth projections for each option as if the "upgrade" was performed at the first equivalent flight. This will give an idea of the expected growth rate for each vehicle. In total, 9 vehicle configurations can be generated from the baseline, upper stage, and booster options in the tables above.

First, the number of flights required to reach a 95% probability of meeting or exceeding the 90% reliability requirement for first flight can be examined for each vehicle. Table 29 lists the vehicles by upper stage and booster type and gives the number of flights required to reach this requirement. Note that the data in the table does not take into account the block upgrade strategy. It only shows the number of required flights for each vehicle if that specific configuration was developed and tested from the start of the program. As seen in the table the baseline vehicle architecture performs much better when considering this requirement. However, this vehicle is expected to be the least capable in terms of performance.

Table 29: Upper stage block upgrade options for SLS

Booster Type	**Upper Stage Type**	**Flights Required**
Baseline (SRB)	Baseline (2 RL-10C1)	70
Baseline (SRB)	4 RL-10C2	136
Baseline (SRB)	2 J-2X	96
Adv. SRB	Baseline (2 RL-10C1)	86
Adv. SRB	4 RL-10C2	152
Adv. SRB	2 J-2X	126
LRB	Baseline (2 RL-10C1)	136
LRB	4 RL-10C2	200
LRB	2 J-2X	168

In order to implement and test the block upgrade strategy using the CONTRAST method, a few additional assumptions will be needed regarding the development schedule of the vehicles. The first such assumption is in regard to the assumed first flight of the baseline vehicle. As shown in the table above, the baseline vehicle reaches the 95% probability of meeting or exceeding 90% reliability at equivalent flight 70. This value will therefore be used as the assumed first full flight of the SLS baseline vehicle. This means that 70 equivalent flights worth of testing is performed on the vehicle subsystems prior to the first full scale test flight. The STS development timeline, discussed in Section 3.11.1, showed between 20 and 100 equivalent flights worth of testing for the vehicle subsystems. Seventy equivalent flights worth of testing therefore represents a very conservative approach in which the subsystems are tested very extensively prior to first flight.

Next, assumptions regarding the development timeline are required. These assumptions will address the starting point of the development for each of the block upgrade options as well as the planned points of upgrade for the vehicle. For the development start dates, multiple different strategies can be identified.

The most desired strategy would obviously be the concurrent development of all the systems. In this strategy all of the upper stage options and booster options would begin development at equivalent flight 0 along with the baseline vehicle. Unfortunately, this approach is the least realistic because budgetary constraints will not allow for all vehicles to be developed concurrently.

A more realistic approach for development of the upper stage and booster options would be to stagger the start dates of the unique development programs. In this case, one of the upgrade options would begin development at some time after the baseline vehicle. After some number of equivalent flights the next upgrade option would begin development. The points at which the upgrades occur could then be determined in two different ways.

First, the block upgrades can be set based upon a specific equivalent flight number. Since the first flight of the baseline vehicle is set at equivalent flight 70, the first block upgrade would be assumed to occur after an additional number of flights. The second upgrade would then occur at a set number of flights after the first upgrade.

The next option for the upgrade timing is based upon the achieved reliability of the upgrade options. In this case the reliability of each option would be observed at each step in time after the first baseline flight. When one of the options reached a specified threshold, an upgrade would be performed. The second upgrade would occur when the other option reached the reliability threshold.

The SLS block upgrade study will implement both of these approaches so they can be compared side by side. It is expected that an interesting trade will be identified between the expected vehicle reliability and the number of flights required to reach the final upgraded vehicle. In the cases where the set upgrade schedule is used, the vehicle reliability at each upgrade may be lower than the other option. However, the upgrades will likely be performed much earlier than if a reliability threshold is implemented for the upgrade options. Thus the vehicle will achieve its highest performance capability earlier in the flight history.

From the discussion of the schedule the following assumptions were derived for the SLS block upgrade study. First, two block upgrade approaches will be used. One will assume a set number of flights in between upgrades, with the first upgrade occurring 10 equivalent flights after the first baseline test flight. The second upgrade will then be assumed to occur 20 equivalent flights after the first upgrade.

The other option will implement a reliability threshold to determine the point at which each upgrade occurs. This threshold will be based upon the requirement setup in the previous section. Each upper stage and booster option will be considered eligible for upgrade when they achieve a probability of at least 95% of meeting or exceeding a reliability of 0.99. This means that a new upper stage or booster will not be placed on the operational vehicle until there is at least 95% confidence that its individual probability of LOV is 1 in 100 flights.

Next, the development timeline of the upgrade options will be staggered. Budgetary constraints make it very unlikely that the baseline, advanced upper stage, and advanced booster would all be developed concurrently. Therefore, a staggered approach is assumed for this study. Since the first flight is set at equivalent flight 70, the time between the start of individual developments will be set at 25. This means the baseline vehicle will start development at flight 0, one of the upgrade options will start at 25, and the last upgrade option will begin development at flight 50. The 25 flights assumption was derived from the notional timeline in Table 18 in Section 3.11.1.

This table shows differences between the assumed first flights of multiple vehicle subsystems that were derived from the STS development timeline. The first flight differences in the table range between 10-80 flights, with an average around 35 flights between components. Since the SLS first flight was assumed at 70, this average was adjusted downward to 25 in order to accommodate the start of development of both upgrade options prior to the first baseline flight.

Using the assumptions derived above a run matrix can be generated, which identifies the development and upgrade strategy combinations. Note that two options exist for both the upper stage and booster upgrades. All combinations of the upper stage and booster options will be used with each case in the run matrix, resulting in 24 unique scenarios. Table 30 below lists the various run cases for the block upgrade study.

Table 30: Run matrix for SLS block upgrade study

Development Order	Upgrade Strategy	First Upgrade
Upper Stage, Booster	Fixed	Upper Stage
Upper Stage, Booster	Fixed	Booster
Booster, Upper Stage	Fixed	Upper Stage
Booster, Upper Stage	Fixed	Booster
Upper Stage, Booster	Threshold	Variable
Booster, Upper Stage	Threshold	Variable

In order to compare the various run scenarios, multiple different outputs will be tracked. The primary outputs to track will be the flight number and reliability distribution for the full vehicle at each block upgrade. The first four cases in the run matrix will have identical flight numbers for the upgrades, however, their reliability distributions will vary depending upon the system being upgraded. The final two cases will have variable flight numbers at each upgrade.

The distributions at each upgrade will ultimately help to determine the best strategy for upgrading the vehicle in terms of reliability. At each upgrade a very minimal decrease in the vehicle reliability is desired, with an increase in vehicle reliability being a major plus. The goal for the vehicle is to reach its final state in the smallest number of flights and at the greatest reliability. Therefore, the reliability distribution at the second block upgrade will be of particular interest because it represents the most evolved and capable version of the vehicle.

In addition to looking at the distributions at each upgrade, the mature reliability distributions for each case can be examined. The mature distributions help to identify the vehicles that will achieve the highest reliability. For this study the mature reliability distributions for each vehicle will be used to test the probability of meeting a reliability requirement of 1 in 100 flights, which was the estimated probability of

LOV for the STS at the end of its operational life.

To begin the discussion of the results, the fixed upgrade strategy will be analyzed first. Recall that this strategy used a set flight number for both of the block upgrades. The order of the development and eventual upgrade of the boosters and the upper stage were varied, which is shown in the run matrix in Table 30. The beginning point of the development of these systems was set to either 25 or 50 flights. Note that the number of repetitions performed at each step in time was set to 1000 in order to keep the runtime at a reasonable level. These repetitions were run at a total of 375 steps in time for each of the vehicles within the run matrix. The data therefore represents a total of 3 million individual repetitions.

First, the reliability growth curves for the fixed upgrade strategy can be examined. Figure 68 shows the mean reliability of all of the vehicles versus equivalent flight number. In the figure the upper stage upgrade types have been colored red and blue, while the booster upgrade types are marked using circles and pluses. The leftmost line of points between flights 0 and 80 represents the reliability growth of the baseline configuration. At flight 80 the first upgrade occurs, which generates four small groupings of points. A few of these groupings can be seen at the end of the arrows marked A and B. After flight 100 the second upgrade occurs, from which the reliability growth of the final vehicle configuration proceeds. This figure shows a few interesting trends associated with the various booster and upper stage options.

First, the booster type in the fixed upgrade strategy seems to have a large effect on the mature reliability of the vehicle. The two lines at the bottom of the plot between flight 100 and 400 (right arrow labeled C) represent all of the liquid rocket booster cases. The other two lines above (left arrow labeled C) show the advanced booster cases, which reach a higher reliability at maturity. The difference between these reliability values is on the order of 1 in 250 flights.

It is also interesting to note that the order of development has a significant impact on the mature reliability when the advanced SRB is used, but does not have much effect when an LRB is used. This is shown by the arrows labeled C. The left arrow points to two growth tracks for the RL-10 upper stage with an advanced booster. As seen in the figure, there is a noticeable split between these tracks during the middle range of the flight history. The uppermost track represents the case where the upper stage begins development prior to the booster, while the lower track represents the opposite. In looking at the right arrow labeled C, both of these tracks lie on top of one another, showing no preference for the development order. This result suggests that in the fixed upgrade strategy, the vehicle reliability will be much more sensitive to the development order if an advanced SRB is used. In the case of the LRB the development order does not make a large difference in the mature reliability. However, it does show a difference when considering the first upgrade.

Two arrows labeled A can be seen pointing to two of the J-2X growth tracks between flight 80 and 100. These tracks represent the case where a single upgrade has occurred. The top-most arrow shows a grouping of points that are cases in which the booster was developed and upgraded first. Note that this grouping contains both circles and pluses, which means that both booster types lay within this group. The lower group of points denoted by A shows the case where the upper stage began development prior to the booster.

The growth tracks labeled B show a similar trend. The top-most track within this grouping represents the case where the upper stage development began first and the upper stage was upgraded first. The lower track therefore represents the case where the booster was developed first but the upper stage was upgraded first. These groupings show a rather intuitive result, which shows that the option that is planned to be upgraded first should also begin development first in order to reduce the effects of the upgrade on the overall vehicle reliability.

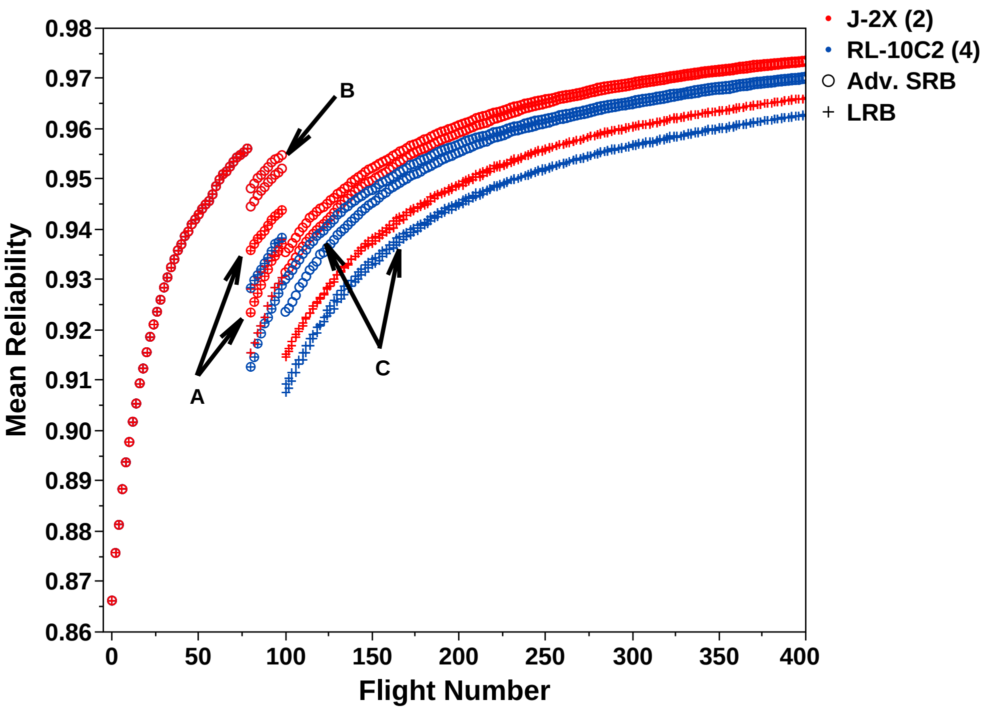


Figure 68: Reliability growth for a fixed block upgrade schedule

Next, the flexible upgrade strategy can be considered in order to identify any differences from the fixed approach. The first flexible approach used a comparison between the mean vehicle reliability and the booster and upper stage upgrade options in order to determine when an upgrade would take place. In this case, the mean of each of the upgrade options was tracked. When the mean of one or both of the upgrade options became greater than or equal to that of the current full vehicle, an upgrade was said to take place.

In this approach the booster upgrade occurred first in every single case, including cases where the booster development was started after upper stage development. The upper stage upgrade was performed simultaneously in all but two cases using this approach. In these two cases the upper stage upgrade was performed at flight 82 and flight 84. Since the upgrades both occurred on or very near flight 80, the results for this strategy show nearly identical trends as the fixed upgrade strategy. The second flexible upgrade strategy therefore presents more interesting results.

The second flexible strategy used a different criterion for determination of upgrade eligibility. In this approach a probability of reaching a specific reliability level was used to determine if the upper stage or booster was ready for upgrade. As discussed above, the initial criterion was a 95% probability of meeting or exceeding a reliability of 0.99 for the individual upper stage or booster. To check this criterion the reliability distribution for each upgrade option at each step in time was sampled. If the percentage of samples exceeding a reliability of 0.99 was greater than or equal to 95%, the associated upper stage or booster was deemed fit for upgrade.

Upon initial run of the 95% criterion it very quickly became clear that this setting was over constraining the block upgrades. Using the 95% setting resulted in almost zero total upgrades, with only the advanced boosters being upgraded at the very end of the flight history. Therefore, the 95% criterion was relaxed until both the booster and upper stage upgrades occurred at some point in the flight history for every case. The resulting criterion was a 50% probability of meeting or exceeding the reliability requirement of 0.99 for the individual upper stage or booster.

The initial results for the second flexible upgrade strategy can be seen in Figure 69, which plots the mean reliability of each vehicle versus equivalent flight number. This plot quickly shows a very different reliability growth outcome for the vehicles when using the new strategy. A vertical line was added to the figure to show the assumed first flight of the baseline architecture. After this reference line each of the upgrade options were eligible to be upgraded if the reliability criterion was met. The arrows labeled A show small splits between a few of the growth tracks that are very similar to the fixed upgrade output. These splits show the effects of development order on the overall vehicle reliability. The three arrows labeled B highlight a few of the discrete jumps in reliability caused by the block upgrades. Since a variable strategy was applied these upgrades occurred at different points throughout the flight history.

The reliability growth tracks for the different vehicle options after the first upgrade are not as tightly grouped as the fixed strategy. At the center of the figure a very long track exists, starting from the top arrow labeled A at flight 80 to flight 450, which reaches the highest vehicle reliability requirement. In this case an advanced SRB booster upgrade was implemented very early in the flight history. Since the upper stage upgrade did not occur until flight 450, the vehicle achieved significant reliability growth prior to this flight. After the upper stage upgrade a significant drop in reliability occurred and both the cases following this growth track ended at nearly the same mature reliability. It is interesting to note that both these cases utilize the RL-10C2 upper stage, which explains why the initial booster upgrade reaches such a high reliability. The RL-10C2 upper stage requires more equivalent flights to reach a mature state, which is ultimately why the advanced booster vehicle progressed so far prior to upper stage upgrade.

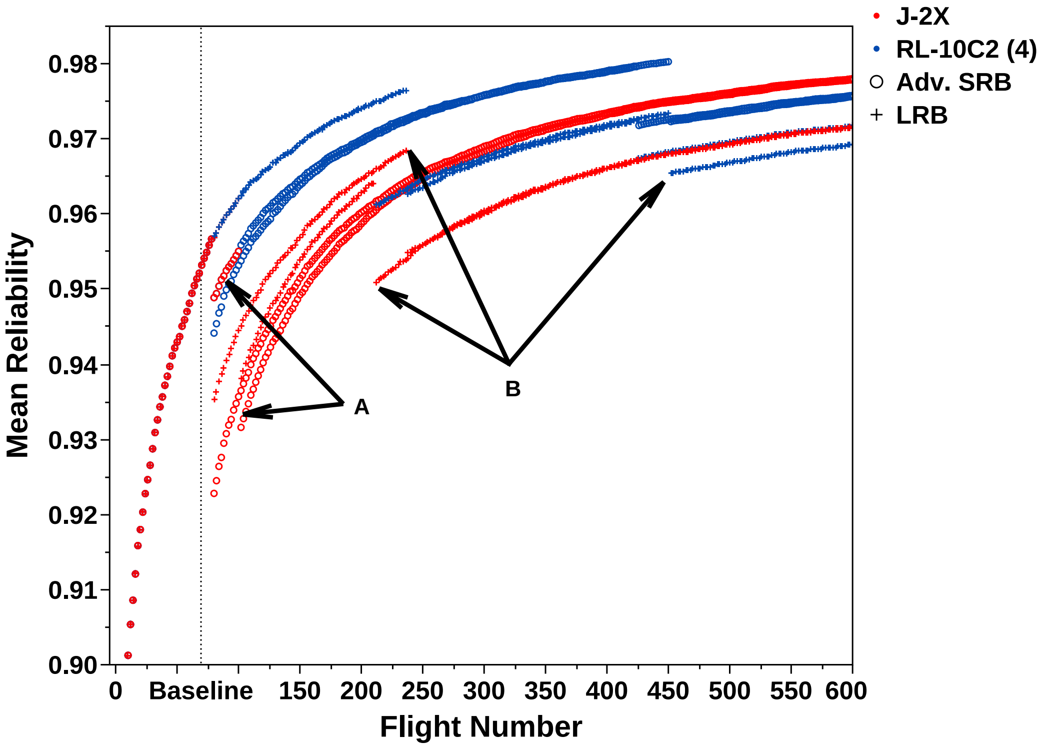


Figure 69: Reliability growth for a flexible block upgrade schedule

Another interesting view of the flexible upgrade strategy data can be created by highlighting the reliability growth tracks based upon the vehicle type. This will show the effects of the various upgrade options on the overall reliability growth track of the vehicle. Figure 70 and Figure 71 show two pairs of cases highlighted. In Figure 70 the pair of selected vehicles includes the advanced solid booster and RL-10C2 upper stage upgrades. Figure 71 shows the advanced solid booster and J-2X upper stage upgrades.

The first major difference to note between the two sets of cases is the point at which the upper stages are upgraded. The RL-10C2 stage requires more time to reach the upgrade criterion, which results in the very long history of the booster upgraded vehicle. In the case of the J-2X stage, there is little to no flight history for the middle vehicle block.

Since the RL-10C2 stage requires more equivalent flights to reach the upgrade requirement, the upper stage upgrade occurs at flight 426 and 452, which is labeled B in Figure 70. This slight difference in flights is caused by the first development setting for the upgrade scenario. The flight number at upper stage upgrade was lower in the cases where upper stage development began prior to booster development.

It is interesting to note that the booster upgrade occurred at flight 80 for both of these cases. This shows that the equivalent flight at booster upgrade was independent of when the booster development began. Thus for the advanced booster and RL-10C2 vehicle it would be advisable to begin development of the upper stage first in order to achieve the final vehicle upgrade at an earlier equivalent flight.

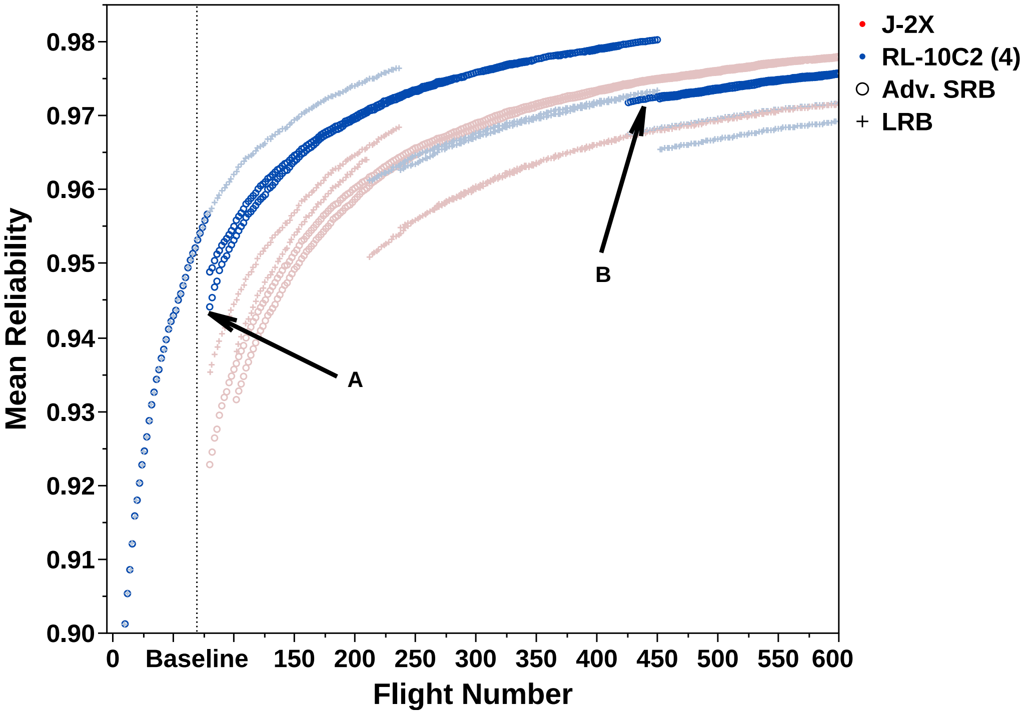


Figure 70: Advanced solid booster and RL-10C2 upper stage vehicles

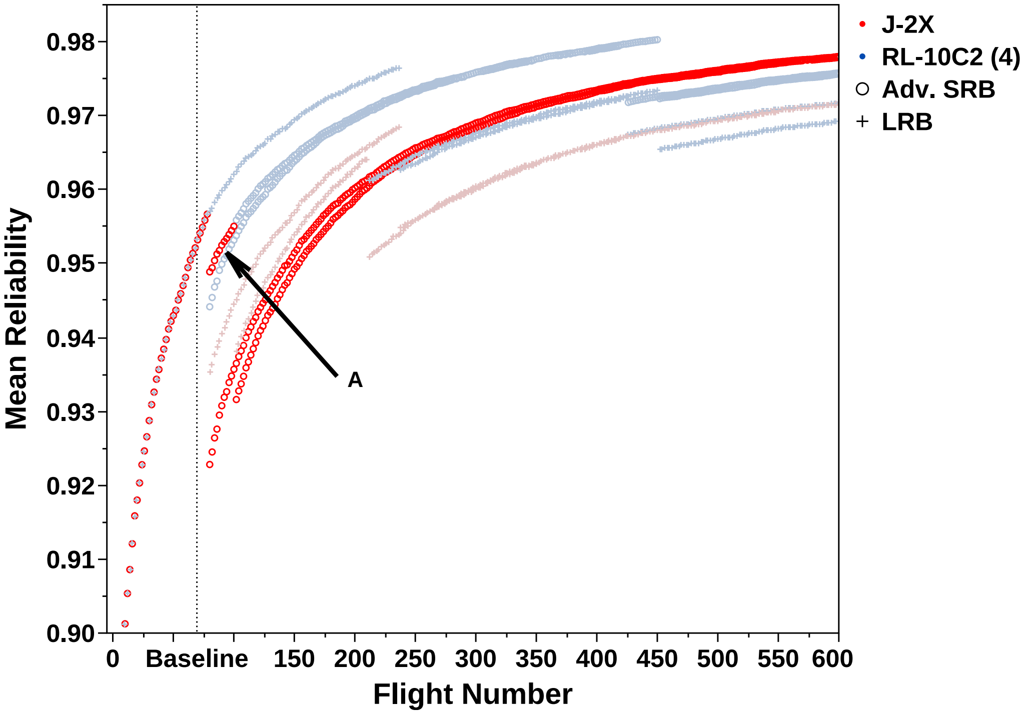


Figure 71: Advanced solid booster and J-2X upper stage vehicles

A similar effect can be seen in the J-2X cases in Figure 71. For this vehicle upgrade case the J-2X upper stage upgrade occurred at flight 80 and 102. The earlier upgrade again occurred when the upper stage development was assumed to start first. As with the RL-10C2 upper stage case, the advanced booster was upgraded at flight 80 for both of the J-2X cases. Therefore, when the J-2X development was assumed to begin prior to the booster development, both upgrades were ready for implementation at the same equivalent flight. Again, this case shows that it may be beneficial to begin upper stage development first in order to reach the fully upgraded vehicle in the least number of equivalent flights. However, the case in which both upgrades occurred simultaneously shows the largest reduction in vehicle reliability from the baseline configuration. If reliability is the only consideration the group of points labeled A in Figure 71 is much more desirable. These points represent the case where the booster was developed and upgraded first.

Another difference between the two plots in Figure 70 and Figure 71 is seen in the mean reliability of the vehicle across the entire flight history. From equivalent flight 0 to flight 80 these values are the same because the baseline vehicle is in use. However, from flight 80 to the end of the flight history the vehicles have much different mean reliabilities. Except for the points labeled A in Figure 71, the advanced booster and RL-10C2 case shows a much higher reliability between flight 80 and 426.

From a pure reliability standpoint this vehicle is much more desirable because it maintains a higher reliability for over half of the flight history. However, the lower expected reliability from the J-2X case is due to the fact that both upgrades have occurred by flight 102. This highlights a major trade between performance and expected reliability. The fully upgraded vehicle will obviously provide more performance between flights 102 and 426, but it also has noticeably lower mean reliability value.

Using the same purely reliability standpoint the most desirable vehicle is different when considering flights above 452. Past this equivalent flight the RL-10C2 upper stage upgrades have occurred, which causes a drop in expected reliability of the overall vehicle. This drop causes the RL-10C2 vehicle to lie below the J-2X vehicle for the remainder of the flight history. In this case, the J-2X vehicle is more desirable in terms of pure reliability. That may also be the case when considering performance, since the J-2X is a higher thrust engine and may be able to push a higher payload mass.

Next, the reliability growth tracks for the liquid rocket booster cases can be considered. Figure 72 and Figure 73 below give two additional highlighted vehicle cases. Figure 72 shows the LRB and RL-10C2 vehicles, while 73 shows the other upgrade combination with the LRB and J-2X upper stage.

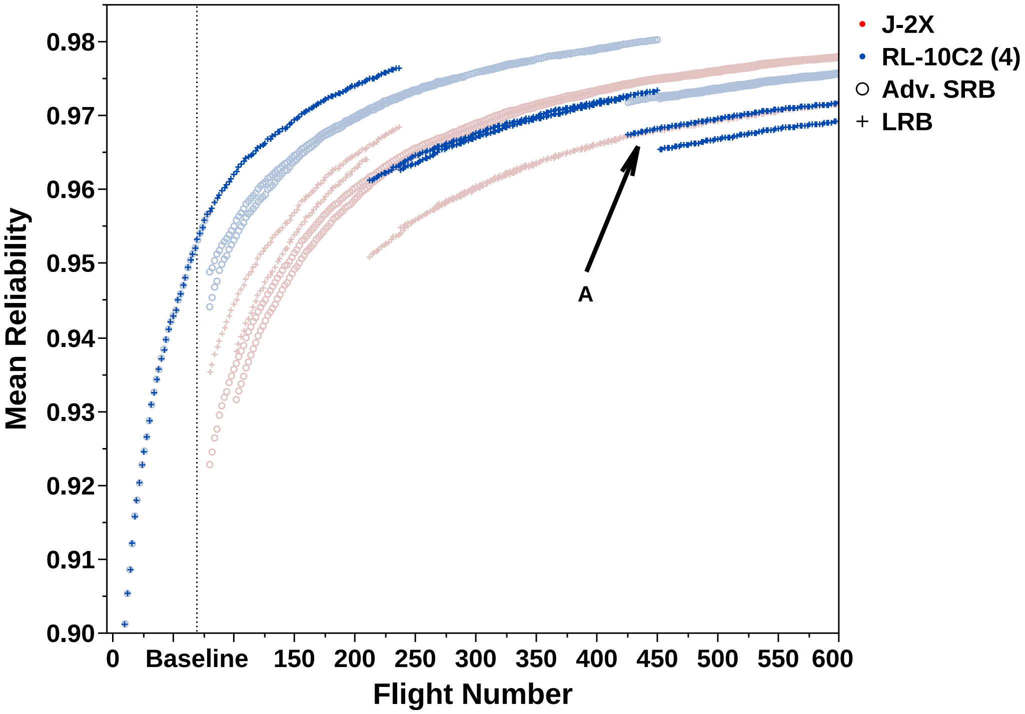


Figure 72: Liquid rocket booster and RL-10C2 upper stage vehicles

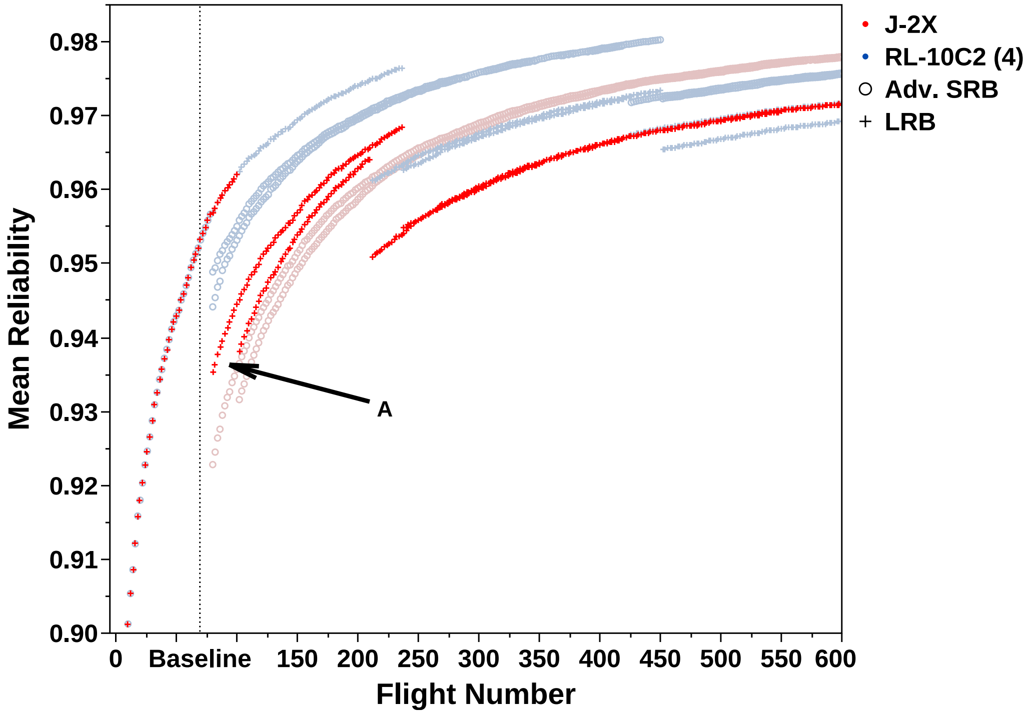


Figure 73: Liquid rocket booster and J-2X upper stage vehicles

For the LRB set of vehicles a difference can now be seen in the flight number at which the first upgrade occurs. The previous case with the advanced solid boosters showed a first upgrade at equivalent flight 80 for every combination of development order and upper stage type. The LRB cases, however, show a different first upgrade point depending upon the upper stage option. In Figure 72 the first upgrade occurs at flight 212 or 238 depending upon the development order of the booster and upper stage. Both of the first upgrades for the RL-10C2 upper stage case are the liquid booster.

The J-2X upper stage cases are much different in terms of the first upgrade. For these vehicle cases the first upgrade occurs at flight 80 or 102 depending upon the development order. Both of the first upgrades for this vehicle are the upper stage. Since the J-2X stage reaches the upgrade criterion more quickly, this vehicle also reaches the full upgraded status in fewer equivalent flights. The full upgraded status is reached at flight 212 or 238 versus flight 426 or 452 for the RL-10C2 vehicles.

It is interesting to note that the development order effects are seen at different points in the flight history for the RL-10 and J-2X upper stage options. In both figures split growth tracks caused by the development order are labeled A. The split shown in Figure 73 is very intuitive. Since the upper stage upgrade occurs first, the top track represents the case where the upper stage was developed first. The only difference between cases after the booster upgrade is the point at which the second upgrade occurred.

The split caused by the development order occurs after the second upgrade in RL-10C2 upper stage cases. This split is labeled A in Figure 72. The lower track corresponds to the case in which the upper stage development begins later than the liquid booster. The upper track therefore corresponds to the case where the upper stage is the first to be developed. These two cases show a noticeable split due to the differences in the growth rates between the liquid booster and the RL-10C2 upper stage.

Since the lower track corresponds to the case where the booster was developed first, the booster reliability for this track is expected to be slightly higher than the upper track. However, in the lower track the upper stage was developed later and will have a lower reliability than in the upper track. What this illustrates is that the reduction in reliability of the upper stage is much greater than the increase in reliability of the booster due to development order. At the later portions of the flight history, the liquid booster has reached a mature state, where the reliability is nearly constant. Due to this, the changes in development time will not have a large effect on the booster reliability. On the other hand, the upper stage is still experiencing some noticeable reliability growth during the later equivalent flights. This means that the extra 25 flights worth of development time will have a significant impact on the stage reliability. Therefore, this split is another illustration of the importance of the order of development for the liquid booster and RL-10C2 upgrade options.

The final analysis that can be performed for the flexible block upgrade data is to consider the reliability growth tracks versus a performance metric. This will allow for the identification of obvious trades that exist between the overall vehicle performance and reliability. It will also illustrate how the CONTRAST method can be integrated with other vehicle analyses in order to make more informed design decisions during early conceptual design.

In order to observe the relationships between reliability and performance for the block upgrade problem a notional payload capability metric was introduced. Development of a full scale modeling and simulation environment that can handle many unique vehicle concepts was considered to be outside the scope of this research. Therefore, the payload capability metric was setup using representative levels of performance, which were based upon the different upgrade blocks of the vehicle.

Figure 67 shows a notional block upgrade scheme for the SLS, which first implements an advanced booster upgrade followed by an upper stage upgrade. At each upgrade the payload capability, listed above each vehicle, is incrementally increased. This trend was used to set up the payload metric that was used herein.

First, the baseline vehicle was assumed to have the lowest payload capability. An upgrade to the baseline upper stage or booster would increase the payload capability to the next level. Based upon the upgrades shown in Figure 67 it was assumed that a booster upgrade would increase the baseline vehicle performance more than an upper stage upgrade. Thus the second performance level corresponds to vehicles with baseline boosters and upgraded upper stages. The third level therefore represents a vehicle with a baseline upper stage and upgraded boosters.

Two additional levels were added to account for the second upgrade of the vehicle. It was assumed that the liquid rocket boosters would provide the best performance in terms of payload capability. Therefore, the highest payload capability level was assigned to vehicles with an upgraded upper stage and liquid rocket boosters. The fourth level was then assigned to the vehicles with advanced solid boosters and upgraded upper stage. A notional 1 to 5 scale was used for illustration purposes, with 5 corresponding to the best payload performance. This scale is representative of a range of payloads such as the 70 ton to 130 ton range seen in Figure 67.

After applying the payload capability metric, the reliability growth data was colored according to the notional scale. Figure 74 below shows the reliability growth tracks colored on a scale from blue to red, with red corresponding to the highest payload capability. Multiple observations regarding trades between reliability and performance can be drawn from this figure.

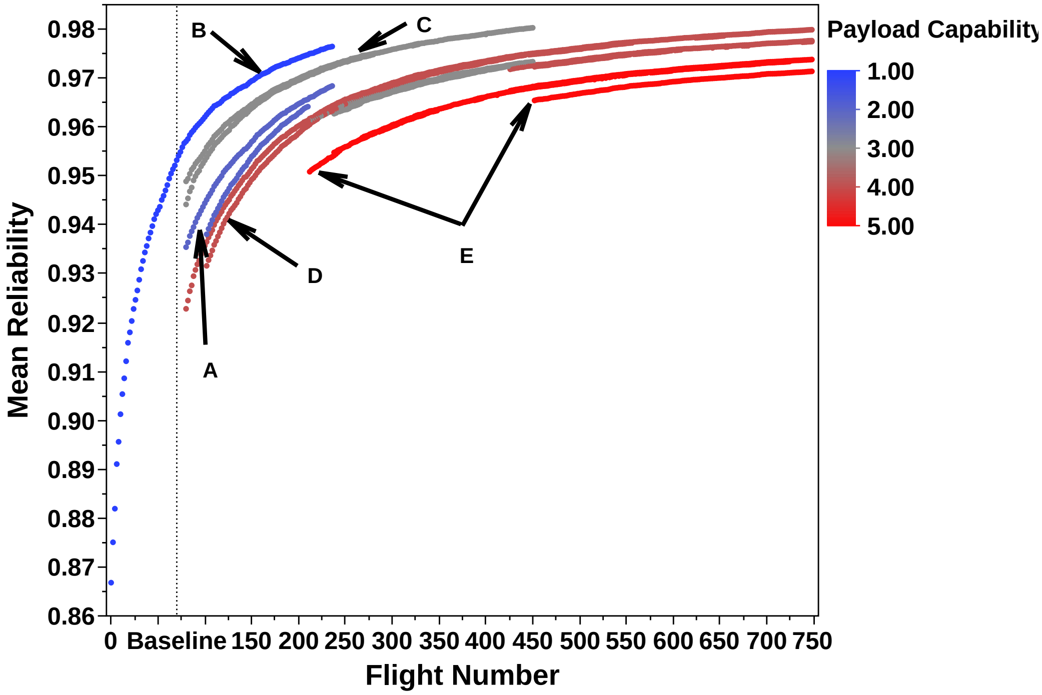


Figure 74: Reliability growth for a flexible block upgrade schedule

First, the effects of the order of upgrade on reliability and performance are apparent in the figure. Consider the cases in light blue between flight 80 and 250 starting from the arrow labeled A. These cases represent vehicles in which the upper stage was upgraded before the booster. In terms of reliability these cases are dominated by the baseline vehicle line of cases in blue, labeled B, as well as the gray line of cases representing a booster upgrade, labeled C. The cases are also dominated in terms of performance as the red line below, labeled D, represents a fully upgraded vehicle. This shows that if the upper stage upgrade occurs first, the vehicle will incur a larger decrease in reliability for a smaller increase in performance than if the booster were upgraded.

However, the early upper stage upgrade does allow the vehicle to reach its highest level of payload capability earlier on in the flight history. The gray line of cases between flight 80 and 450, labeled C, shows a better performance and reliability for a vehicle on which the booster was upgraded first. Unfortunately for this vehicle, the second upgrade occurs very late in the flight history around flight 450. In contrast, the light blue cases labeled A, where the upper stage was upgraded first, reach the second upgrade point between flight 200 and 250. There are also cases in which the booster and upper stage were upgraded simultaneously, causing a jump in capability from 1 to 4 around flight 80.

This data ultimately shows a trade-off that depends upon the difference in capability between the upper stage upgraded vehicle and the booster upgraded vehicle. If a significant payload capability increase can be achieved with a booster upgrade, then the difference in time required to reach the final maximum payload capability may be acceptable. However, if this difference is negligible the upper stage upgrade first approach looks more appealing. This is due to the fact that the vehicles that upgraded the upper stage first reached the maximum payload capability in a much smaller number of equivalent flights.

A similar trade-off can also be seen when considering the two booster options. Since the liquid booster was assumed to provide more performance, the level 5 points in red all contain liquid boosters and upgraded upper stages. The flights at which performance level 5 is reached are labeled E. The final vehicle configurations using advanced solids were set to level 4, which is labeled D and shown in light red. As seen in the figure the liquid rocket booster vehicles lie beneath the advanced solid cases in terms of reliability. The liquid vehicles also are at a disadvantage in terms of time to reach the final payload capability. At the bottom of the figure the lighter red points begin at flight 80, while the red points don't show up until around flight 210. In this case it is again important to consider the differences between the booster capabilities. If the liquid booster provides a significant advantage in terms of performance, the added number of required equivalent flights may be acceptable. However, if the performance difference is small or the schedule and budget are prohibitive the advanced solid booster is the more desirable option.

The consideration of the trade-off between reliability and time required to reach the final configuration is heavily dependent upon the total lifespan of the vehicle. Obviously 200 operational launches is a very large and unrealistic expectation for a single launch vehicle. However, the equivalent flights take into account any testing that occurs in between operational flights. For example, subsystems such as the SSME and SRB on the STS underwent test firing between every operational flight. The desired upgrade option is therefore dependent upon the assumed schedule and approach for testing between flights. If a rigorous post flight analysis and test approach is implemented, the vehicle may reach equivalent flight 200 very rapidly. In this case, the LRB's lower reliability between flight 100 and 200 is less of an issue because the final vehicle configuration will be reached very rapidly. On the other hand, if testing between flights is prohibited by an aggressive launch schedule or budget, reaching equivalent flight 200 will take much longer if it is reached at all. This case makes the cases labeled D more attractive because they reach the final vehicle configuration at a much earlier point in the flight history.

5.3.2.5 Conclusions

The three sections above illustrated multiple different uses for the output from the conceptual reliability growth method. Section 5.3.2.1 covered some basic data exploration techniques for initial analysis of the reliability growth output. This section illustrated a few techniques to rapidly identify the expected effects of the architecture options. By using simple filtering the architecture options that seem to have the largest effects on reliability can be quickly identified.

The next section, Section 5.3.2.2, went into further detailed analysis of the reliability growth results. In this section the probabilities of meeting or exceeding a reliability requirement were derived for each vehicle within the architecture space. An example was also given, which compared the output from the CONTRAST method to the output of one of the state-of-the-art tools discussed in Section 2.3. This example illustrated the ability of the method to improve upon one of the specific weaknesses that was called out for the existing tool.

Section 5.3.2.3 then presented an alternative approach to what was seen in Section 5.3.2.2. In this section the number of flights required to reach a threshold for the first operational flight was explored. Eight example vehicles were used to illustrate the effects of various architecture options on the time required to reach the first flight threshold. In addition, prediction profilers were introduced as an alternative method for identifying the effects of each parameter. These profilers give the analyst a rapid and intuitive visualization of the architecture effects.

The final results section, Section 5.3.2.4, presented a more complex application of the CONTRAST method. This section addressed the proposed block upgrades to the SLS vehicle and looked at the effects of the various upgrade options on vehicle reliability and time required to reach the final vehicle configuration. The upper stage and booster upgrade options for this section were setup based upon proposed upgrades to the SLS baseline vehicle. Multiple upgrade strategies were also implemented.

The results of the block upgrade study ultimately conveyed the importance of the development order in terms of vehicle reliability. The results showed that the upper stage development should be started prior to the booster development especially when using the advanced solid booster.

The upgrade order was also shown to have an impact on the equivalent flights required to reach the final vehicle configuration. In general, the vehicles which upgraded the upper stages first also reached the final configuration first. The time required to reach this final configuration is very important when considering the payload capability of the upgraded vehicles. It also has very major implications when considering the available budget and schedule. A large number of required equivalent flights will obviously require many expensive tests as well as time to perform said tests.

Within the final results section a notional payload capability metric was also added to the results in order to illustrate trades between reliability and performance. Although full performance analysis was not implemented for all of the vehicles, Figure 74 was used to show how such results could be used to further supplement the output of the CONTRAST method. The joint consideration of performance and reliability with a single set of results was one of the primary motivating factors for the initiation of this research. Utilizing the method derived in this dissertation along with established vehicle performance analysis tools will allow for more informed decision making during early conceptual design. Ultimately this will translate into more effective concept down selection, which will result in baseline concepts that find a balance between maximum performance and maximum reliability.

CHAPTER VI

CONCLUDING REMARKS

6.1 Summary of Findings

The primary goal of this thesis was to develop a method for assessment of launch vehicle reliability and safety during conceptual design. It was shown in Section 1.1 that vehicle architecture options have a great effect on the eventual reliability of the system during operations. However, during the early phases of the design cycle, vehicle options are down-selected and the baseline architecture is essentially locked-in. Therefore, it is essential that the architecture effects on reliability and safety be captured prior to the selection of a baseline vehicle. This observation lead to the derivation of the research objective for the thesis, which is restated below.

Research Objective

To formulate and implement a method that will quantitatively capture launch vehicle architecture effects on reliability and safety, in order to facilitate more informed decision making during early conceptual design.

In order to meet the research objective multiple research questions were derived in Chapter 3. These questions ultimately guided the research and helped to determine the appropriate steps for the final method. The first research question, posed in Section 3.2, addressed the desired form of the final output. This question acted as the starting point for development of the method, working backwards from the desired output. The desired output format was identified using derived requirements for research objective completion. In reality only a few options for the output format exist, which made estimates as a function of time a logical choice. This choice was reflected in the statement of an assertion to research question 1.

Assertion to Research Question 1

Reliability estimates as a function of time are the most desirable for comparison of launch vehicle concepts during early design because they provide more information than point or probabilistic estimates.

Following the selection of estimates as a function of time as the desired output for the method, Section 3.3 continued on to the second research question. Prior to this research question reliability growth methods were identified as a natural selection for generating the method output. In Section 3.3 the options for potential growth models were reduced to consider only discrete formulations, which are more appropriate for assessment of launch vehicles. Therefore, research question 2 was posed in order to identify the most appropriate discrete reliability growth model for application during conceptual design.

Section 3.4 carried out experimentation in order to select a growth model for application in the CONTRAST method. In this section, the model options were further reduced after discussion of the data availability during early conceptual design. Models such as the Fries and Finkelstein models were eliminated because they required data that is not likely to be available early in the design cycle. Ultimately Experiment 1 illustrated that the Hall model captured the reliability growth behavior of previous vehicles much better than the Morse model. The results of this experiment therefore substantiated hypothesis 2.

Hypothesis 2

If the Hall growth model is applied, the output reliability estimates will have greater accuracy while utilizing more traceable assumptions than the AMSAA-Crow, Finkelstein, Fries, or Morse growth models.

The next step in the development of the method stemmed from a weakness in the application of the reliability growth models. It was noted that if these models are applied at the system level, various architecture effects cannot be captured. Thus, trying to compare vehicles that are nearly identical but only introduce changes to subsystem redundancy or engine-out is futile if the growth model is applied at the system level. Research question 3 was posed in order to address this issue and identify the appropriate level of application of the growth models.

In Section 3.5 various levels of system characterization were discussed and compared to the flow through the design process. It was observed that details regarding each level of characterization are essentially unlocked as the design progresses. This means that during early design, only information regarding the top most levels is available. However, it was noted that the growth model accuracy was expected to increase as the level of characterization became more detailed. Therefore the desired level of characterization for model application was identified as the lowest level at which data was available. At this point another observation was made, which further reduced the options for the level of application of the growth models.

The assumptions for the selected Hall growth model were discussed in Section 3.3.1.5. Within the derivation of the model the failure modes are assumed to occur independently of one another. In other words, the failure modes included in the model cannot cause the initiation of another failure mode within the model. The easiest way to avoid this issue is to only define the catastrophic failure modes. However, at the more detailed levels of characterization it is much more difficult to identify independent modes. At levels such as the assembly or sub-assembly, multiple strings of failures need to take place in order to cause an LOM or LOV event. Therefore, the options for model application were limited to the system and subsystem level in order to avoid an issue with failure mode independence.

The level of application of the Hall model was then addressed by research question 3. During the discussion of research question 3 it was shown that the lowest levels of application were inappropriate for use during conceptual design. This was primarily due to the amount of information required to produce the model assumptions. The highest level of application was also ruled out. Applying the model at the system level would not allow for the capture of some primary architecture options for improving reliability such as redundancy or engine-out capability. An assertion to research question 3 was stated, which selected the subsystem level as the most advantageous during conceptual design.

Assertion to Research Question 3

Applying the reliability growth models at the subsystem level will provide an adequate level of detail to capture relevant architecture trades while avoiding issues with data availability.

After identifying the subsystem level as the most appropriate for application of the reliability growth model a new research question was posed. This question addresses the need for an additional step within the CONTRAST method to take the subsystem level reliability growth outputs and generate an overall vehicle growth curve. Therefore, research question 4 asked what technique is most appropriate for generating the vehicle level reliability growth curve using the subsystem level growth models.

In Section 3.6 five options were identified as possible solutions. Through the discussion in this section multiple options were eliminated from consideration, leaving only two for testing in Experiment 3. The two remaining options, FTA and RBD, were expected to perform exactly the same for any given launch vehicle architectures. The results of the experiment eventually substantiated this expectation, showing no difference between the two options. Therefore, the choice between the remaining options was determined to be a matter of preference. A simple fault tree was then identified as the approach for combining the subsystem level growth curves. Fault trees were chosen because they are failure oriented, which aligns well with the failure mode based assumptions of the growth model.

Two additional research questions were posed in order to work out details regarding the CONTRAST method. The first of these questions examined the options for generating the reliability growth assumptions. Through the discussion of this question in Section 3.9 multiple potential avenues were identified for assumption generation. It was determined that the assumption sources would vary depending upon the amount and quality of the available data for each subsystem. Both a simple parts count approach and a detailed approach based upon FMEA were discussed. These approaches were eventually demonstrated in the STS example problem in Section 4.2.

Within the example problem an important observation was drawn regarding the approaches for generating the growth model assumptions. Although the parts count approach is very rudimentary, it was successfully applied during the example problem for multiple subsystems. The SSME subsystem within the example problem benefited from the availability of detailed FMEA data, which gave accurate estimates of number of failure modes and probability of occurrence. For this subsystem a simple parts count approach was taken first and then compared to the detailed SSME FMEA data. The resulting comparison showed agreement between the parts count derived assumptions and the detailed FMEA. This effectively showed the validity of the parts count approach in the absence of detailed data. The final reliability growth projections that were generated using these assumptions also showed very satisfactory agreement with the actual STS data. The results ultimately supported the assertion that was developed in response to research question 5.

Assertion to Research Question 5

- If detailed data from a previous, similar system exists, then the number of failure modes and probability of occurrence assumptions can be generated based upon this data with the fix effectiveness factors coming from SME input.
- If detailed data from a previous, similar system does not exist, then the number of failure modes assumption can be rapidly estimated using the PCM approach and the probability of occurrence distribution must be assumed to represent a generic complex system. The fix effectiveness factors can be generated using SME input.

The final research question addressed the application of the subsystem level reliability growth curves to a fault tree. Multiple different approaches were identified for generating the overall vehicle reliability growth curve. These approaches were primarily concerned with how the subsystem reliability growth curves should be incremented in relation to the overall vehicle. These approaches were tested in Experiment 4 in Section 3.11, which ran each approach on a set of representative vehicle architectures. The results of this experiment showed that an anchored approach was most appropriate because it did not require additional data for setup and provided the greatest accuracy. Experiment 4 therefore substantiated hypothesis 6 from Section 3.10.

Hypothesis 6

If all subsystem growth curves are anchored at equivalent flight 0, the resulting system level growth curve can be produced without increasing evaluation time or encountering data availability issues while maintaining an acceptable prediction accuracy.

After completion of the experiments and example problem an application problem was presented in Chapter 5. It was designed to be a demonstration of the CONTRAST method on a relevant launch vehicle design problem. The SLS heavy lift launch vehicle was chosen as an excellent example for the application problem because it is currently progressing through the design cycle. In addition, multiple block upgrade strategies for the upper stage and boosters are being considered today.

Based upon the SLS vehicle architecture a matrix of alternatives was developed. The goal of the matrix was to capture relevant vehicle architecture options for an SLS-like vehicle as well as include novel concepts such as a winged fly-back booster. The final matrix of alternatives is presented in Figure 39 in Section 5.1.1.

The results of the application problem were used to show the benefits of applying the CONTRAST method for reliability assessment during conceptual design. First, basic data exploration was demonstrated, followed by more detailed analysis. Within the detailed analyses a primary comparison was made to one of the state-of-the-art reliability tools, FIRST, which was discussed during the literature review. This comparison was performed in order to illustrate one of the primary weaknesses of the state-of-the-art tool and to show how the CONTRAST method can successfully improve upon the area of weakness.

The final section within the application problem results illustrated a unique capability of the CONTRAST method. The block upgrade approach for the SLS vehicle was investigated further in this section using the reliability growth output. Multiple upper stage and booster upgrade options were included, which allowed for the analysis of various upgrade strategies. The application of the method allowed for the exploration of trades between time required to reach maturity and overall probability of meeting a specific reliability target. Trades between reliability and performance were also demonstrated within this section. The block upgrade study ultimately showed the versatility of the hybrid reliability growth approach and illustrated the additional analyses that are enabled by this method.

6.2 *Summary of Contributions*

The work presented in this thesis provided multiple contributions to the field of reliability assessment during conceptual design. These contributions relate to the specific techniques applied within the method as well as the successful completion of the research objective itself.

The first contribution is related to the application and associated assumptions of reliability growth models at the subsystem level. In Section 3.5 the discussion of research question 3 identified the weaknesses in applying growth models at the system level. Within this discussion it was concluded that the weaknesses could be alleviated by applying the growth model at a lower level of characterization. However, the change in level of application also required an appropriate approach for model assumption generation. During the discussion of research question 5 approaches for developing these assumptions were explored. Ultimately two approaches were demonstrated during the example problem, which served as a validation exercise of the CONTRAST method. The example problem showed the validity of utilizing a parts count approach in the absence of detailed data in order to generate an accurate reliability growth projection. It also illustrated the connection between the parts count based assumptions and actual incremental upgrades that may occur throughout the life-cycle. The results from the experiments and example problem provided the groundwork for a simple, traceable, flexible, and accurate approach for generating reliability projections during early design.

The second contribution of this research is the demonstrated application of reliability growth projections to a system level fault tree. During the discussion of research question 3, a primary weakness in applying reliability growth models was identified. This weakness resulted in the inability to capture basic changes in vehicle architecture, which was one of the desired characteristics of the new method. The contribution in this area relates to the improvement of the capture of vehicle architecture effects when using reliability growth projections. Development of the model assumptions at the subsystem level versus the system level was demonstrated during the example problem. By implementing a hybrid fault tree and growth model approach one of the primary weaknesses of the growth models has been alleviated. Therefore, this approach successfully enables the comparison of "unique but similar" vehicles during conceptual design.

The final contribution of this research is the hybrid reliability growth approach itself. The primary objective of this research was to produce a method that is capable of providing the analyst with more knowledge about the expected vehicle reliability and safety during conceptual design. During the development of the primary objective, three requirements were derived that would signify the successful completion of the research objective. These requirements can be considered as a sort of "wish list" for the new reliability approach and represent areas of improvement beyond currently available techniques.

First, the method developed within this thesis provides a very flexible approach for assessing reliability and safety during early design. Using a matrix of alternatives to automatically generate vehicle level fault trees allows the analyst to assess a very wide range of vehicle concepts in a very short period of time. As long as the appropriate reliability growth assumptions are generated and stored within the matrix, heritage and completely new vehicle concepts can be evaluated within the same architecture space. The use of a matrix of alternatives based storage approach for the reliability assumptions also provides a high level of re-usability. Assumptions from previous studies can be simply ported over to a new matrix of alternatives.

Second, the resulting reliability method provides a much needed link between the conceptual design team and the reliability analysis team. Typically, these teams carry out their tasks with little to no interaction during the early phases of design. In some cases, reliability analysis is not even considered as essential for conceptual design studies. With the method developed in this thesis the resulting reliability analyses from the reliability team can now be incorporated into the performance analyses performed by the design team. As shown in the application problem, the results from this method are well suited for visualizing trades between performance and reliability, which can help the design team in concept down selection.

Due to the fact that the method can operate with very simple or very detailed reliability assumptions, the link between the reliability group and design group does not need to be prohibitive. During an actual study the heritage subsystems within the matrix of alternatives will most likely benefit from the availability of detailed reliability information. However, this detailed information and knowledge does not lie within the design team, but with the reliability team. Therefore a database type approach can be taken when developing the reliability growth assumptions for various subsystems.

For example, vast amounts of test data and reliability analyses were generated for the SSME as it proceeded from initial development through operations. At each point in this process, the reliability team can use the knowledge gained to generate very accurate reliability growth assumptions for an SSME-like liquid rocket engine. With the assumptions now stored, the design team can query the information during a future design study that may consider the SSME as an engine option for a core stage. Using this approach would help reduce the amount of information that the design team needs to request from the reliability analysts. Thus, the only information that the design team would need to request is in regard to novel concepts that have not been explored in previous studies. The hybrid reliability growth method has therefore provided an approach that allows the experts in both the reliability and performance analysis fields to coalesce their respective results into one coherent package.

Finally, the method developed within this thesis provides a flexible, traceable, and accurate approach for assessing vehicle architecture effects on reliability prior to baseline selection. As discussed throughout the introduction to this research, the selection of a baseline vehicle can have very large consequences on the eventual reliability. It is therefore imperative that the effects of various architecture options be quantified and utilized during the concept selection phase of the design. A few existing tools were identified that can support this task, however, weaknesses were identified in each. Based upon these weaknesses a method was desired that would produce quantitative reliability predictions with the flexibility to incorporate novel concepts. In addition the method needed to perform rapid analyses of the vehicles in order to support the exploration of very large architecture spaces. Unlike the existing techniques, the hybrid method resulting from this thesis has been shown to possess all of these desired characteristics. The resulting method can therefore facilitate more informed decision making during early conceptual design by incorporating reliability and safety as a figure of merit for vehicle down selection.

6.3 Recommendations for Further Research

While conducting research as a part of this thesis multiple different avenues for additional research have been identified. These avenues pertain to specific areas within the CONTRAST method as well as the exploration of alternative concepts for use of the method. First, the method developed within this thesis would benefit from additional research into the generation of the distributions for probability of occurrence.

The probability of occurrence distributions are used for each of the subsystems within the matrix of alternatives of the CONTRAST method. They can be derived using generic complex system assumptions or from reliability data for the system itself. However, in the case of heritage hardware a common pitfall can be encountered. Typically heritage hardware is considered to be more reliable due to the fact that it has been flown successfully before. Therefore a common misconception is that if the heritage hardware is slapped on a new vehicle, the vehicle will automatically become more reliable. The problem becomes more apparent after returning to the definition for reliability, which states "for a given period of time under specified conditions". The key words here are "specified conditions". Although the heritage hardware may have demonstrated a 100% success rate in the past, the conditions the hardware may see on a new vehicle will be different. Therefore, the reliability of the equipment may actually decrease when it is implemented on a new vehicle!

With this in mind, further research into the effects of utilizing heritage hardware in a new environment will benefit the CONTRAST method. An approach for adjusting the reliability growth assumptions can potentially be developed and included in the current method. This approach would allow for the adjustment of not only the number of failure modes but also the probability of occurrence assumptions within the growth models.

Another area for further research would be to link this analysis to cost and schedule estimates for the vehicle concepts. Cost and schedule are two additional parameters that are very important to the eventual success of a launch vehicle program. The reliability growth models used within the CONTRAST method are very well suited for cost and schedule analysis as well. During the example and application problems in this thesis a generic equivalent flight metric was used to measure time. The number of flights required by each vehicle to reach a specific reliability requirement or maturity will have a very large effect on the number of required tests, sets of hardware, and time in terms of weeks or months.

For example, consider the case in which a vehicle is expected to reach a specified reliability requirement in 50 equivalent flights. These equivalent flights may be accomplished via full scale testing, however, unsuccessful tests can cause schedule delays. In the event that a test has failed, post-test reporting and design correction are required, which can stretch the test schedule. If no failures are encountered the 50 equivalent flights worth of testing may take on the order of weeks. However, if multiple failures are encountered the total test time may require months to years. The CONTRAST method would therefore benefit from a translation between the equivalent flight scale and actual time (i.e. months, years). This translation would help calculate the expected time to completion of the equivalent flights and identify concepts that are high risk in terms of development time and cost.

A final area for potential research is to study the implementation of more complex fault trees or the augmentation of the current trees using analyses such as stochastic Petri nets. As noted in the review of Petri nets in Section 3.2.1.9, SPN is particularly well suited for analyzing situations in which the failure rate changes in time. This situation typically arises in launch vehicles that possess an engine out capability. If a single benign engine failure occurs, the probability of failure for the other engines will likely increase because they will either throttle up or operate for a longer period of time. Thus, for the analysis of engine out capability it may be possible to introduce simple SPN analysis within the current FTA framework. Instead of implementing the engine out FTA equations as done in this thesis an SPN could be queried, which would generate a more accurate probability of failure for the remaining engines. However, this approach would require appropriate connections between the reliability growth models and the SPN in order to ensure that the correct initial probabilities of failure were being utilized.

APPENDIX A

CLASS STRUCTURES FOR GENERATION OF FTA AND RBD EQUATIONS

```
class MOArow:
    def _init_(self,name,n_opt,opts,r_type,rltn,dep,lnks,r_num):
        self.name = name
        self.num = n_opt
        self.options_list = opts
        self.row_type = r_type
        self.relationship = rltn
        self.dependency = dep
        self.link = lnks
        self.rnum = r_num
    def selection(self,i):
        return self.options_list[i]


class Component:
    def __init__(self,name,rel,ccf,num_pts,num_steps,firstFlight):
        self.name = name
        self.rel = rel
        self.ccf = ccf
        self.curStep = 0
        self.MeanRel = 0
        self.firstStep = firstFlight
        self.fmList = []
        self.parmList = []
        self.relArray = np.zeros((num_pts,num_steps))
        self.relList = []

    def MeanRelCalc(self,step):
        self.MeanRel = np.mean(self.relArray[:,step])

    def HallReliabilityGrowth(self,n_steps,n_pts,n_flts):
        self.relArray = HallGrowth(n_flts,n_steps,n_pts,self.fmList)

    def ReliabilityCalc(self,step,n_reps):
        parms = self.parmList[step]
        a=parms[0]
        b=parms[1]
        self.relList = ss.beta.rvs(a,b,loc=0,scale=1,size=n_reps)

    def RelUpdate(self,nrep,nstep):
        self.rel = self.relArray[nrep,nstep]

    def fmListClear(self):
        #Clears the fmList for the given component
        self.fmList = []

    def fmListAppend(self,fmObj):
        #Appends a new entry into the failure mode list
        n = fmObj.nmodes
        name = fmObj.name
        comp_name = fmObj.component
        number = 1
```

```
alpha = fmObj.alpha
beta = fmObj.beta
fef1 = fmObj.fef1
fef2 = fmObj.fef2
dic={}
for i in range(0,n):
    new_name = str(name) + str(i)
    dic[i]=FailureMode(newname,comp,num,alpha,beta,fef1,fef2)
    self.fmList.append(dic[i])
```

APPENDIX B

HALL GROWTH MODEL IMPLEMENTATION IN PYTHON

```
def HallGrowth(num_flights,num_steps, num_points, fmList):
        import numpy as np

    #Set up the probabilities of failure
    for f in fmList:
        #Set the initial probabilities of failure
        f.BetaRV(num_points)
        #Set the indicator function
        f.IndicatorFn(num_points,num_flights)
        #Set the fix effectiveness factors
        f.FixEffectiveness(num_points)

    #Initialize array
    #Rows are each trial, cols are steps in time
    fit_data=np.zeros((num_points,num_steps))

    #Loop through number of repetitions
    for r in range(0,num_points):
        #Initialize arrays
        RofT_array=np.zeros((1,num_steps))
        #Loop through number of steps
        for n in range(0,num_steps):
```

```
        #Now loop through the failure modes
        RofT=1.0
        #Correct the step number to reflect flight number
        fl_per_step = num_flights/num_steps
        cur_num = n*fl_per_step
        #Set indicator function for all modes
        for f in fmList:
            firstfail = f.IndList[r]
            if firstfail <= cur_num:
                ind=1
            else:
                ind=0
            fef = f.fefList[r]
            p_i = f.betaList[r]
            inner_val = (1-(1-ind*fef)*p_i)
            RofT = RofT*inner_val
        #RofT is rel for this step and rep
        RofT_array[0,n] = RofT
    #Append the new point to data list
    fit_data[r,:]=RofT_array
```

APPENDIX C

DERIVATION PROCEDURES FOR PROBABILITY OF OCCURRENCE ASSUMPTIONS

The purpose of this appendix is to outline the procedures used to derive the probability of occurrence assumptions for the Hall growth model. These procedures were used throughout the experiments and example problem in Chapter 4 and the application problem in Chapter 5. Three different approaches exist for deriving the probability of occurrence distribution depending upon the data that is available for the subsystem being modeled. Each approach will be outlined in the subsequent sections.

C.0.1 No Subsystem Reliability Data

The first approach is the simplest as it deals with the complete lack of data regarding the subsystem. This case will primarily occur when considering novel concepts that have not been developed before. To begin the first approach the subsystem level probability of failure distribution must be assumed. This distribution takes the place of any reliability estimates or data that would normally be available for a heritage subsystem.

As discussed in Section 3.4.1 and 4.2, in the event that no reliability data is available, a general form of the probability of failure distribution can be assumed. Both of the reliability growth models that were considered in Experiment 1 developed similar probability of failure distributions for generic complex systems. This distribution, Beta(0.22,8.75), was used during the example problem in Section 4.2. The example problem results showed an agreement between the model results and the actual reliability growth data for the STS vehicle. The generic distribution for complex systems was therefore deemed acceptable for use in the absence of actual data. Thus, the first step of this approach is to assume the subsystem probability of failure is distributed as Beta(0.22,8.75).

From the subsystem probability of failure the probability of occurrence distribution for the failure modes can be estimated. First, the number of failure modes assumption is needed for the subsystem being considered. With the number of modes, N, known the probability of failure of the subsystem can be written:

$$P_{SubsystemFailure} = P_{Mode1} + P_{Mode2} + ... + P_{ModeN} \tag{30}$$

In this equation, $P_{SubsystemFailure}$ is distributed as Beta(0.22,8.75) and the failure mode probabilities $P_{Mode1}, ..., P_{ModeN}$ are drawn from the probability of occurrence distribution, $Beta(\alpha, \beta)$. A simple optimization algorithm can then be used to determine the alpha and beta parameters for the probability of occurrence distribution such that the resulting sum approximates the probability of subsystem failure distribution. In this case the optimization is carried out until the distribution resulting from the summation of the probabilities of occurrence matches the mean, standard deviation, minimum, and maximum values of the subsystem failure distribution.

C.0.2 Single Point Reliability Estimate

The second approach involves the case where a reliability estimate is available for the subsystem under consideration. In this case the reliability estimate is in the form of a single point such as 0.987 or 0.99. Reliability data such as this are relatively easy to obtain because they can be directly calculated using a basic flight history. For example, the STS solid rocket booster flew on 135 total missions with only 1 catastrophic failure. The flight history suggests that the demonstrated reliability of the booster was 0.996 or 1 failure out of 270 total boosters. This is a very crude estimate for the expected reliability of the subsystem, however, all that is needed is a starting point on which to anchor the reliability growth models.

With a single point estimate for the subsystem, multiple approaches can be taken to derive a probability of occurrence distribution for the failure modes. The point estimate can be assumed as the mean, maximum, or minimum starting reliability for the subsystem. For the application problem in Chapter 5 any point estimates were assumed as the mean reliability for the subsystem. It is suggested that the value be set to the maximum expected reliability for subsystems that will be used in a new environment or that have not been operated for an extended period of time. For example, one of the liquid booster engine options in the application problem was an F-1 derived gas generator. If point estimate of the F-1 reliability were to be used as a surrogate for the new gas generator engine, setting this estimate to the maximum reliability for the new engine is most appropriate. This is due to the fact that the F-1 engine has not been operated since the early 1970's. The operating environment of the new gas generator will also be completely different than the F-1 engine on the Saturn V.

After determining the mean value for the subsystem reliability the variance can either be set based upon expert judgment or a confidence bound can be calculated using the flight history. For the example and application problems the variance was set based upon the reliability estimate and demonstrated reliability of the subsystem. For example, the RS-68 engine had a predicted reliability from [175] of 0.9987, but throughout its flight history on the Delta-IV vehicle no major failures have occurred. In this case the variance was set based on the difference between these two values. Ultimately, a larger variance value is desired in order to represent the rather large amount of uncertainty associated with the point estimates.

Alternatively, the variance value can be estimated using procedures for developing confidence bounds. In this case, the reliability point estimate can be considered as a binomial parameter, p. The flight history of the subsystem is then made up of successes and failures that are distributed as $B(n, p)$, where n is the total number of trials and p is the probability of success. The variance of p can then be written as $var(p) = \frac{p(1-p)}{n}$ [150]. An approximate confidence interval could also be produced for p using simple equations presented in [2]. Given X successes in n trials:

$$\tilde{n} = n + z^2 \tag{31}$$

$$\tilde{p} = \frac{1}{\tilde{n}}(X + \frac{1}{2}z^2) \tag{32}$$

then the confidence interval for p is,

$$p \pm z\sqrt{\frac{1}{\tilde{n}}\tilde{p}(1-\tilde{p})} \tag{33}$$

where $(1-\alpha)$ is the level of confidence and $z = 1 - \frac{1}{2}\alpha$.

After the mean and variance parameters for the subsystem probability of failure have been determined, the parameters for the associated Beta distribution can be directly calculated. First, assuming a mean of μ and variance of σ^2 [98]:

$$\mu = \frac{\alpha}{\alpha+\beta} \tag{34}$$

$$\sigma^2 = \frac{\alpha\beta}{(\alpha+\beta^2)(\alpha+\beta+1)} \tag{35}$$

Using Equations 34 and 35 the distribution parameters can then be written as a function of the mean and variance:

$$\alpha = \left[\frac{1-\mu}{\sigma^2} - \frac{1}{\mu}\right]\mu^2 \tag{36}$$

$$\beta = \alpha\left[\frac{1}{\mu} - 1\right] \tag{37}$$

Following the development of the Beta parameters for the subsystem level probability of failure distribution the procedures are exactly the same as in case 1. Using Equation 30 and a simple optimization procedure, the Beta parameters for the failure mode probability of occurrence distribution can be approximated.

C.0.3 Reliability Data Given with Confidence Bounds

The final approach for generating the failure mode probability of occurrence distributions refers to the case where more detailed reliability data is available for the subsystem under consideration. During the application problem this was the case for the reliability of the RS-25 and J-2 engines. Data from reference [103] was used for each of these engines, which gave a mean reliability value along with a 5^{th} and 95^{th} percentile value.

To generate the subsystem reliability distribution from this data a simple function was coded in Python, which solved for the Beta parameters. In the function, shown below, the optimizer adjusts the settings for the Beta parameters in order to minimize the objective function. The objective function in this case was defined as the square of the error between the given data and the current distribution. Depending upon the percentile metrics given by the actual data, the objective function can be adjusted to include additional requirements. During the application problem, the maximum value was also included in the objective function as the scale parameter for the Beta distribution. In the code shown below, x1 and x2 refer to the reliability values for the i^{th} and j^{th} percentiles, while p1 and p2 define the specific percentile (i.e. 95^{th}). As with case 2, following the generation of the subsystem distribution the probability of occurrence distribution can be derived as in case 1. Equation 30 and an additional optimization are used to approximate the Beta parameters for the failure mode probability of occurrence distribution.

```
def beta_parameters(x1,p1,x2,p2,mean,mx):
    def square(x):
        return x*x
    def objective(v):
        (a, b) = v
        temp=square(stats.beta.cdf(x1,a,b,loc=0,scale=mx)-p1)
        temp+=square(stats.beta.cdf(x2,a,b,loc=0,scale=mx)-p2)
        temp+=square(stats.beta.mean(a,b,loc=0,scale=mx)-mn)
        return temp
    # arbitrary initial guess of (3, 3) for parameters
    xopt = optimize.fmin(objective, (3, 3))
    return (xopt[0], xopt[1])
```

APPENDIX D

SENSITIVITY TO NUMBER OF REPETITIONS

As noted in Chapter 5 the number of repetitions at each step in time is an important consideration for both runtime and model accuracy. With too few points the vehicle reliability distributions at each step in time will be very sparse causing difficulties with model fitting and possible bias in the mean values. Using too many points however, will require a much larger amount runtime, which will reduce the number of architectures that can be analyzed. This study was setup to identify an appropriate range for the number of repetitions setting. Ideally, the number of repetitions used will be large enough to accurately resolve the mean reliability at each step in time while requiring a minimal runtime.

In order to carry out the sensitivity study a representative stage was selected from the application problem. Only one stage versus an entire vehicle was selected in order to reduce the amount of time required to carry out the study. The RL-10C2 stage was chosen for this study because it contains 4 individual components that require separate reliability growth curves. Recall that this stage contains 4 RL-10C2 engines, a redundant avionics system, an IVF power system, and a structures/other subsystem. The number of steps and number of flights from the application problem were reduced in the interest of runtime. Therefore, 250 total flights were run with 50 steps in time.

In order to test a wide range of number of repetition values while keeping the required runtime at an acceptable level, six values were selected on a graduated scale. Starting from the lowest point, 250, the number of repetitions was doubled until reaching 8000. The six settings for number of repetitions therefore were: 250, 500, 1000, 2000, 4000, and 8000. Each of these settings was run for 50 trials in order to produce ranges on the mean reliability values at each step in time. Obviously a setting for number of repetitions is sought that will have a very minimal range of mean values while keeping runtime at an acceptable level.

First, the runtime required for each setting can be seen in Figure 75. This figure shows an exponential type trend, however, note that the number of repetition settings were not generated on a linear scale. The figure illustrates about a 2 times increase in required runtime for the growth models with a 2 times increase in number of repetitions. The increase in model runtime is therefore expected to show a linear trend with number of repetitions. In terms of runtime Figure 75 shows that a number of repetitions below 2000 is preferable, which requires approximately 30 seconds to evaluate the growth model. At 8000 repetitions, the runtime is nearly 2 minutes, which would be extremely prohibitive when evaluating large architecture spaces.

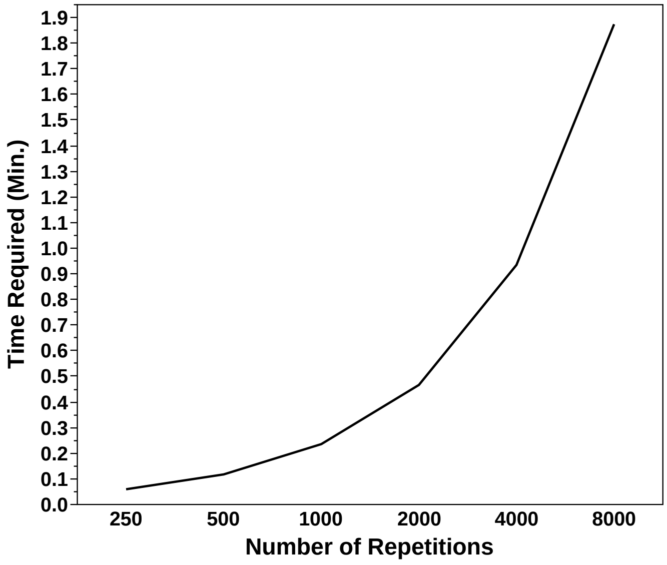
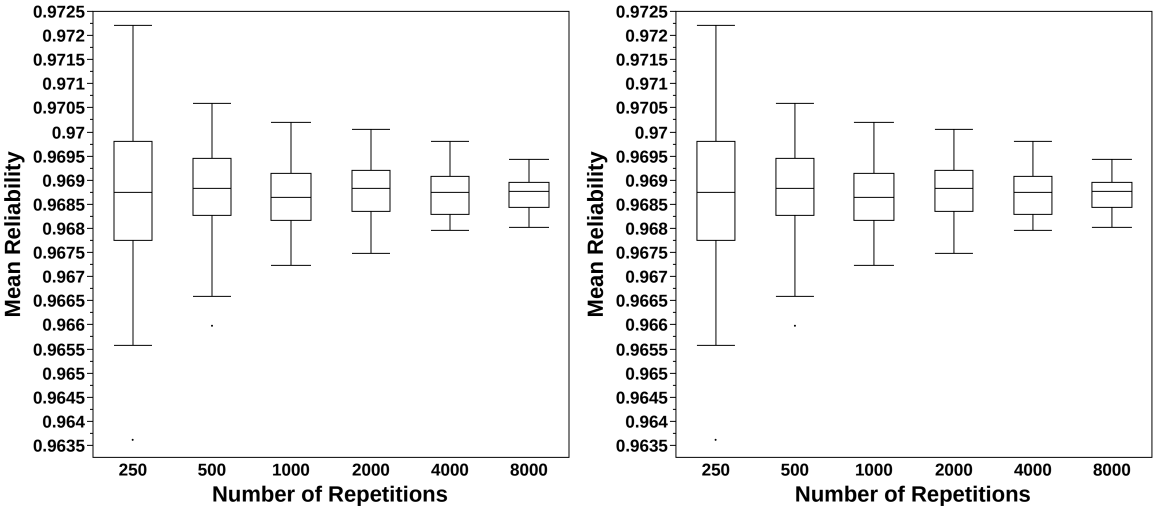


Figure 75: Required reliability growth model runtime for different repetition settings

Next, the distributions of the mean reliability values for each step in time and each repetition setting can be analyzed. Figure 76 below shows box plots at two points in time for all of the repetition settings. These plots illustrate the range in mean values that were produced by the 50 individual runs of the growth models for each setting. Naturally, the range of values decreases as the number of repetitions increases.



(a) Flight 100

(b) Flight 250

Figure 76: Mean reliability box plot at two flights for varying repetition settings

As seen in Figure 76 the range of mean reliability values produced by the 8000 repetitions option is approximately $\frac{1}{5}th$ the range of the 250 repetitions option. There is also a fairly noticeable difference between the ranges of the 250 and 500 repetitions options. At 1000 repetitions the range in mean values is about twice the size of the 8000 repetition option. However, the absolute value of this range is only around 0.033. In comparison, the range at 8000 repetitions is around 0.015. The difference in runtime between these options is nearly an order of magnitude.

From the results produced from the number of repetitions study a generic suggestion for future studies can be offered. The goal of the study was to identify a number of repetitions setting that would keep runtime at a reasonable level and achieve an acceptable convergence on the mean reliability values. For future studies it is suggested that between 1000 and 2000 repetitions be used for the reliability growth models depending upon the overall size of the architecture space.

APPENDIX E

NUMBER OF FAILURE MODES SENSITIVITY STUDY

The final trade study will examine the effects of varying the number of failure modes assumption within the reliability growth model. The goal of this section is to identify changes in the reliability predictions that will occur due to errors in the number of failure modes assumption. This will help give the analyst an idea of how accurate the number of modes assumption must be in order to maintain an acceptable output accuracy.

In order to test the sensitivity to the number of modes assumption a single subsystem will be analyzed. The subsystem selected for this study was the RL-10C2 engine, which was used within the number of repetitions study in Appendix D. This subsystem was run for 250 equivalent flights at step size of 5 flights per step. A total of 2000 repetitions were used for the generation of the reliability distribution of the subsystem at each step in time.

The original setting for the RL-10C2 number of failure modes was derived in Section 5.2.1.1. This assumption was set to 10 total modes for the RL-10C2 in the application problem. In order to test the sensitivity to this assumption, a range about this original setting was applied. For the sensitivity study the reliability growth curve for the RL-10C2 was generated using a number of modes between 5 and 15.

First, the differences in the mean reliability of the engine over its flight history can be examined. Figure 77 below plots the mean reliability for each of the number of modes settings. The curves have been colored by the number of modes with blue representing the lowest value (5) and red for the highest (15).

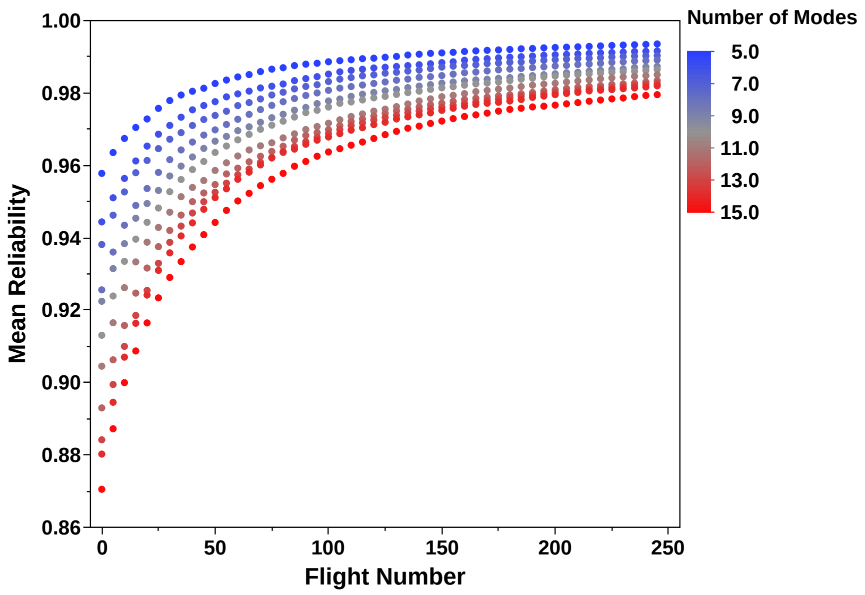


Figure 77: Mean engine reliability with varying number of failure modes

As seen in the figure a very large difference exists between the mean reliability predictions using 5 and 15 failure modes. For the 5 failure mode case, the initial mean value is 0.9577, while the 15 failure mode case has an initial mean value of 0.8705. Fortunately this difference decreases as the equivalent flight number increases. At the final flight the difference between the maximum and minimum failure mode settings is 0.014.

The difference between the minimum and maximum settings for the number of failure modes becomes more significant when considering the distributions throughout the time history. Figure 78 plots the probability of meeting a specified reliability requirement for each setting throughout the flight history. In this case the reliability requirement was set at 0.99, thus the figure shows the probability that the engine has a reliability of at least 0.99 at each step in time. As seen in the figure, the 5 mode case reaches close to 80% at the final flight, while the 15 mode case is around 25%.

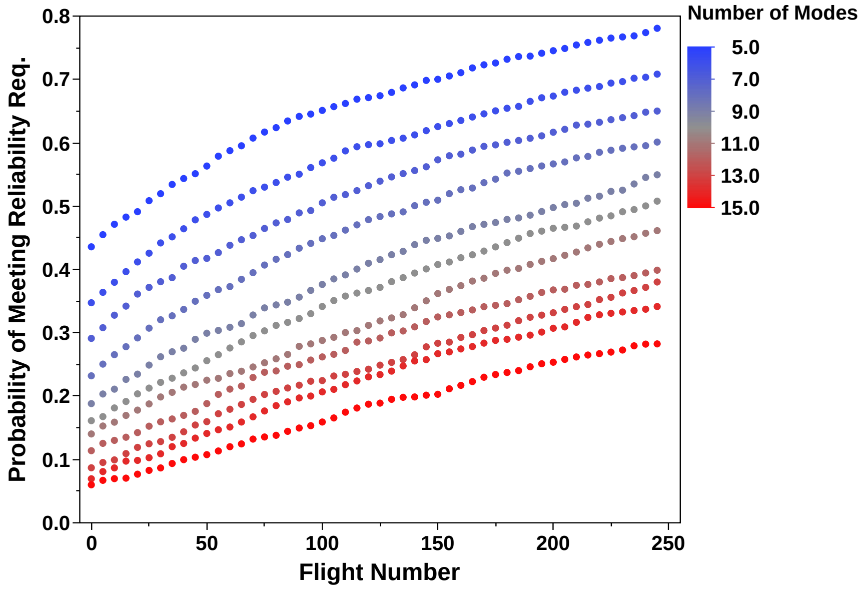


Figure 78: Probability of meeting a reliability requirement with varying number of failure modes

The large differences in the probabilities of meeting the requirement are due to the varied distribution shapes that result from the failure mode settings. Obviously, setting the number of failure modes to 5 will result in a much lower variability in the output than setting the modes to 15. Thus, the ranges of the distributions for higher mode settings are wider, which ultimately reduces the resulting probability of meeting the requirement in Figure 78. Figure 79 shows the reliability distributions at flight 250 for three mode settings. The far left distribution corresponds to 5 failure modes, the center is 10 modes, and the right is 15 modes. This figure shows the large difference in range between the 5 and 15 mode cases.

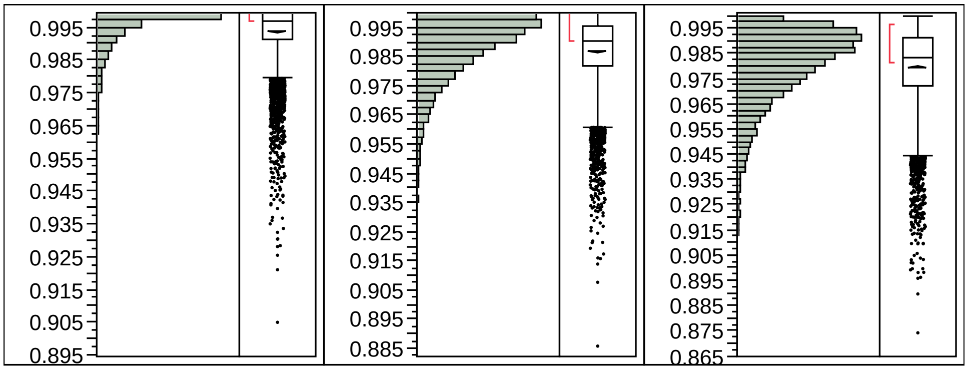


Figure 79: Reliability distributions at flight 250 for three mode settings

The differences seen in Figure 77 and 78 between 5 and 15 modes are relatively large, however, it is important to note that this range represents a ±50% change from the original number of failure modes for the subsystem. If a smaller range of failure modes is considered, the differences seen in the mean reliability values decrease drastically. At ±2 failure modes and flight 250 for example, the difference in mean reliability is 0.00625 and 0.00886 for the lower and higher mode values, respectively. These values are less than 1% above or below the mean value at 10 failure modes.

The primary take-away from this study is the importance of consistent definition of the level of characterization for the subsystems. In carrying out the CONTRAST method it is very important to ensure that all subsystems are considered at the same level of detail in order to avoid gross over or under estimation of the number of failure modes assumption. This is especially true when generating the assumption with a parts count approach. The parts count should only include the parts that reside on the level of characterization of interest. If a part is too detailed or too abstract for the level of interest, errors in the number of failure modes assumption may result. Ultimately, a range of $\pm 2-3$ modes will not be a deal breaker in terms of prediction accuracy. However, in vehicles with many different subsystems special care should be taken to ensure consistent assumption generation across all subsystems.

REFERENCES

[1] Achenbach, J., “NASA budget for 2011 eliminates funds for manned lunar missions.” Washington Post, February 2010.

[2] Agresti, A. and Coull, B., “Approximate is Better than ’Exact’ for Interval Estimation of Binomial Proportions,” *The American Statistician*, vol. 52, pp. 119 – 126, May 1998.

[3] Alliant Techsystems, *ATK Space Propulsion Products Catalog.* August 2012.

[4] Arcidiacono, G., “Development of a FTA versus Parts Count Method Model: Comparative FTA,” *Quality and Reliability Engineering International*, vol. 19, no. 5, pp. 411 – 424, 2003.

[5] Auvation, “OpenFTA Software.” Free download available at: http://www.openfta.com/default.aspx.

[6] Aven, T. and Jensen, U., *Stochastic Models in Reliability.* New York, New York: Springer, 2 ed., 2013.

[7] Basharin, G., Langville, A., and Naumov, V., “The life and work of A.A. Markov,” *Linear Algebra and Its Applications*, vol. 386, pp. 3 – 26, July 2004.

[8] Benton, M., “Reusable, Flyback Liquid Rocket Booster for the Space Shuttle,” *Journal of Spacecraft and Rockets*, vol. 26, no. 4, pp. 252–258, 1989.

[9] Bergin, C., “Second J-2X Engine Prepares for SLS Testing.” Online, February 2013. http://www.nasaspaceflight.com/2013/02/second-j-2x-engine-sls-testing/.

[10] Bergin, C., "The Dark Knights - ATK's Advanced Boosters for SLS Revealed." Online, January 2013. http://www.nasaspaceflight.com/2013/01/the-dark-knights-atks-advanced-booster-revealed-for-sls/.

[11] Biggs, R. E., "Space Shuttle Main Engine The First Ten Years: Part 2 - The Obstacles," *American Astronautical Society History Series*, vol. 13, pp. 69–122, 1992.

[12] Bilstein, R., *Stages to Saturn: A Technological History of the Apollo/Saturn Launch Vehicle*. DIANE Publishing, 1999.

[13] Binkley, J., Clark, J., and Spiekermann, C., "Improved procedure for combining atmospheric flight loads on the day of launch," *Journal of Spacecraft and Rockets*, vol. 37, pp. 459 – 462, August 2000.

[14] Blair, J., Ryan, R., Schutzenhofer, L., and Humphries, W., "Launch vehicle design process: Characterization, technical integration, and lessons learned," tech. rep., National Aeronautics and Space Administration, May 2001.

[15] Blischke, W. and Murthy, D., *Reliability Modeling, Prediction, and Optimization*. New York, NY: Wiley, 2000.

[16] Bloomer, L., "Reliability Assessment of Conceptual Launch Vehicles," tech. rep., NASA Faculty Fellowship Program, Huntsville, AL, 2004.

[17] Blum, C., Jones, P., and Meinders, B., "Dual Liquid Flyback Booster for the Space Shuttle," tech. rep., Lockheed Martin Space Systems, New Orleans, LA, 1998.

[18] Boyer, R., "Space Shuttle Probabilistic Risk Assessment (SPRA) Iteration 3.2," (Cleveland, OH), October 2010.

[19] Broad, W., “For U.S. Satellite Makers, a No-Cost Bailout Bid,” *The New York Times*, April 2009.

[20] Buzzell, J., “Testing for the J-2X Upper Stage Engine,” *SpaceOps Conference*, April 2010.

[21] Campbell, W. and Braun, E., “Reliability - intrinsic or afterthought,” *AIAA 5th Propulsion Joint Specialist Conference*, June 1969.

[22] Cates, G., Gelito, J., Stromgren, C., Cirillo, W., and Goodliff, K., “Launch and assembly reliability analysis for human space exploration missions,” *IEEE Aerospace Conference Proceedings*, 2012.

[23] Chandler, G., Denson, W., Rossi, M., and Wanner, R., “Failure Mode/Mechanism Distributions,” tech. rep., Reliability Analysis Center, Rome, NY, September 1991.

[24] Chang, I.-S., “Investigation of space launch vehicle catastrophic failures,” *Journal of Spacecraft and Rockets*, vol. 33, no. 2, pp. 198 – 205, 1996.

[25] Chang, I.-S., “Space launch vehicle reliability,” *Crosslink*, vol. 2, pp. 23–32, Winter 2001. The Aerospace Press, Los Angeles, CA.

[26] Chang, K., “Telescope behind schedule and over budget, panel says,” *The New York Times*, November 2010.

[27] Chen, A., *Celebrating 30 Years of the Space Shuttle Program*. National Aeronautics and Space Administration, 2012.

[28] Christensen, C., “Launch prices and the economics of the space industry,” *AIAA SPACE 2010 Conference & Exposition*, September 2010.

[29] Clark, S., “Court filings detail sea launch’s bankruptcy,” *Spaceflight Now*, June 2009.

[30] Cohen, H., "Space Reliability Technology: A Historical Perspective," *IEEE Transactions on Reliability*, vol. R-33, no. 1, pp. 36 – 40, 1984.

[31] Corcoran, W., Weingarten, H., and Zehna, P., "Estimating reliability after corrective action," *Management Science*, vol. 10, no. 4, pp. 786 – 795, 1964.

[32] Creech, S., "SLS Dual Use Upper Stage (DUUS) Opportunities," (Huntsville, AL), NASA, April 2013.

[33] Crocker, A., "Update on Risk Reduction Activities for an F-1 based Advanced Booster for NASA's Space Launch System," *50th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, July 2014.

[34] Crocker, A., Doering, K., Cook, S., Bachtel, F., St. Germain, B., and Schaffer, M., "The Benefits of an Advanced Booster Competition for NASA's Space Launch System," *49th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, July 2013.

[35] Crow, L. H., "An improved methodology for reliability growth projection," Tech. Rep. TR-357, AMSAA, Aberdeen Proving Ground, MD, June 1982.

[36] Crow, L. H., "Amsaa discrete reliability growth model," tech. rep., US Army Materiel System Analysis Activity Methodology Office, Aberdeen Proving Ground, 1983.

[37] Crow, L. H., "The extended continuous evaluation reliability growth model," *Proceedings - Annual Reliability and Maintainability Symposium*, 2010.

[38] Cruzen, C., Chavers, G., and Wittenstein, J., "Operational considerations and comparisons of the saturn, space shuttle and ares launch vehicles," *IEEE Aerospace Conference*, 2009.

[39] de Selding, P., "Sea launch rocket fails during liftoff; satellite lost," *NBC News*, February 2013.

[40] Department of Defense, *Military Standard: Procedures for Performing a Failure Mode, Effects and Criticality Analysis*, MIL-STD-1629A ed., November 1980.

[41] Department of Defense, *Military Handbook: Reliability Growth Management*, MIL-HDBK-189 ed., February 1981.

[42] Department of Defense, *Military Standard: Reliability Modeling and Prediction*, MIL-STD-756B ed., November 1981.

[43] Department of Defense, *Military Standard: Reliability Program Requirements for Space and Launch Vehicles*, MIL-STD-1543B ed., October 1988.

[44] Department of Defense, *Military Handbook: Reliability Prediction of Electronic Equipment*, MIL-HDBK-217F ed., December 1991.

[45] Department of Defense, *Department of Defense Standard Practice: System Safety*, MIL-STD-882E ed., May 2012.

[46] Dhillon, B., *Reliability Engineering in System Design and Operation.* New York, NY: Van Nostrand Reinhold, 1983.

[47] Dodson, B. and Nolan, D., *Reliability Engineering Handbook.* Tuscon, AZ: QA Publishing, 1999.

[48] Duane, J., "Learning curve approach to reliability monitoring," *IEEE Transactions on Aerospace*, vol. AS-2, pp. 563 – 566, April 1964.

[49] Dulac, N. and Leveson, N., "Incorporating safety risk in early system architecture trade studies," *Journal of Spacecraft and Rockets*, vol. 46, no. 2, pp. 430 – 437, 2009.

[50] Electric Power Research Institute, "Computer Aided Fault Tree Analysis System (CAFTA) Version 6.0." Free demo available at: http://www.epri.com/abstracts/Pages/ProductAbstract.aspx?ProductId=000000000001015514.

[51] Ellner, P. and Hall, J., "Amsaa maturity projection model based on stein estimation," *Proceedings - Annual Reliability and Maintainability Symposium*, pp. 271 – 277, 2005.

[52] Ellner, P. and Wald, L., "Amsaa maturity projection model," *Proceedings of the Annual Reliability and Maintainability Symposium*, pp. 174 – 181, 1995.

[53] European Cooperation for Space Standardization, Noordwijk, Netherlands, *Space Product Assurance: Safety*, March 2009.

[54] European Cooperation for Space Standardization, Noordwijk, Netherlands, *Space Product Assurance: Components Reliability Data Sources and their use*, 1 ed., January 2011.

[55] European Space Agency, "Space engineering: Requirements and standards," October 2009.

[56] Fabisinski, L. and Maples, D., "Risk evaluation in the pre-phase a conceptual design of spacecraft," *AIAA SPACE 2009 Conference & Exposition*, August 2009.

[57] Finkelstein, J., "Logarithmic reliability-growth model for single-mission systems.," *IEEE Transactions on Reliability*, vol. R-32, no. 5, pp. 508 – 511, 1983.

[58] Fragola, J. and Morse, E. L., "Application of PRA in Development Systems," tech. rep., Valador Inc., January 2012.

[59] Fragola, J. R., "Risk comparison of crew launch vehicle concepts," *Proceedings - Annual Reliability and Maintainability Symposium*, 2012.

[60] Fragola, J., "Safe crew launch by design," *Proceedings - Annual Reliability and Maintainability Symposium*, 2006.

[61] Fragola, J., "How safe must a potential crewed launcher be demonstrated to be before it is crewed?," *Journal of Loss Prevention in the Process Industries*, vol. 22, pp. 657 – 63, September 2009.

[62] Fragola, J., *Aerospace Failure Data Handbook: A Reference Guide to Understanding and Assessing Risk and Reliability of Aerospace Systems and Spacecraft Design*. Herdon, VA: Valador Inc., 2010.

[63] Fragola, J., Booth, L., and Shen, Y., "Current launch vehicle reliability practice and data base assessment," Tech. Rep. AL-TR-89-013, Science Applications International Corporation, June 1989.

[64] Frey, D., Palladino, J., Sullivan, J., and Atherton, M., "Part Count and Design of Robust Systems," *Systems Engineering*, vol. 10, no. 3, pp. 203 – 21, 2007.

[65] Fries, A. and Sen, A., "A survey of discrete reliability-growth models," *IEEE Transactions on Reliability*, vol. 45, no. 4, pp. 582 – 604, 1996.

[66] Fries, A., "Discrete reliability-growth models based on a learning-curve property," *IEEE Transactions on Reliability*, vol. 42, no. 2, pp. 303 – 306, 1993.

[67] Gardner, J., Addis, A., and Marino, A., "Independent Orbiter Assessment: Assessment of Instrumentation Subsystem FMEA/CIL," tech. rep., McDonnell Douglas Astronautics Company, Houston, TX, February 1988.

[68] Gernand, J., Gillespie, A., Monaghan, M., and Cummings, N., “Constellation ground systems launch availability analysis: enhancing highly reliable launch systems design,” *SpaceOps 2010 Conference*, April 2010.

[69] Gillespie, A. and Monaghan, M., “Allocating reliability and maintainability goals to nasa ground systems,” *Proceedings - Annual Reliability and Maintainability Symposium*, 2013.

[70] Glover, R., Kelley, B., and Tischer, A., “Studies and analyses of the space shuttle main engine: Ssme failure data review, diagnostic survey, and ssme diagnostic evaluation,” tech. rep., NASA Marshall Space Flight Center, Huntsville, AL, December 1986.

[71] Go, S., “A Historical Survey with Success and Maturity Estimates of Launch Systems with RL10 Upper Stage Engines,” *Proceedings - Annual Reliability and Maintainability Symposium*, 2008.

[72] Gompertz, B., “On the nature of the function expressive of the law of human mortality, and on a new method of determining the value of life contingencies,” *Transactions of the Royal Society*, June 1825.

[73] Grinstead, C. M. and Snell, J. L., *Introduction to Probability.* American Mathematical Society, 1997.

[74] Hage, R., “A simulation model for probabilistic analysis of space shuttle abort modes,” tech. rep., NASA Marshall Space Flight Center, Huntsville, AL, November 1993.

[75] Hall, J. and Mosleh, A., “A reliability growth projection model for one-shot systems,” *IEEE Transactions on Reliability*, vol. 57, pp. 174 – 181, March 2008.

[76] Hall, J. B. and Mosleh, A., "An analytical framework for reliability growth of one-shot systems," *Reliability Engineering and System Safety*, vol. 93, no. 11, pp. 1751 – 1760, 2008.

[77] Hall, J., *Methodology for Evaluating Reliability Growth Program of Discrete Systems.* PhD thesis, University of Maryland, 2008.

[78] Hamlin, T., Thigpen, E., Kahn, J., and Lo, Y., "Shuttle Risk Progression: Use of the Shuttle Probabilistic Risk Assessment (PRA) to Show Reliability Growth," *AIAA SPACE 2011 Conference & Exposition*, September 2011.

[79] Hamlin, T., "Shuttle risk progression- Focus on historical risk increases," *International Journal of Performability Engineering*, vol. 9, no. 6, pp. 633 – 640, 2013.

[80] Harris, C., "Spacex: Building rockets from the ground up," in *National Council of Space Grant Directors Fall 2012 National Meeting*, (Seattle, WA), 2012.

[81] Hart, D., "The Boeing Company EELV/Delta IV Family," *AIAA Defense and Civil Space Programs Conference and Exhibit*, 1998.

[82] Hertzfeld, H., Williamson, R., and Peter, N., "Launch vehicles: An economic perspective," tech. rep., The George Washington University Space Policy Institute, Washington, D.C., September 2005.

[83] Hinsdale, L., Swain, L., and Barnes, J., "Independent Orbiter Assessment: FMEA/CIL Assessment Final Report," tech. rep., McDonnell Douglas Astronautics Company, Washington, D.C., September 1988.

[84] Huang, Z., Fint, J. A., and Kuck, F. M., "Key reliability drivers of liquid propulsion engines and a reliability model for sensitivity analysis," *41st AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 2005.

[85] Huang, Z. and Weber Jr., T. F., “Liquid propulsion launch vehicle reliability in closed form,” *Journal of Spacecraft and Rockets*, vol. 36, no. 5, pp. 701 – 703, 1999.

[86] Human Exploration Framework Team, “HEFT Phase I Closeout,” National Aeronautics and Space Administration, September 2010.

[87] Isakowitz, S., *International Reference Guide to Space Launch Systems.* AIAA, 3 ed., 2004.

[88] Krevor, Z. C., *A Methodology to Link Cost and Reliability for Launch Vehicle Design.* PhD thesis, Georgia Institute of Technology, Atlanta, Georgia, August 2007.

[89] Kyle, E., “Space Launch Report: Delta IV Data Sheet.” Online, July 2014. http://www.spacelaunchreport.com/delta4.html#delta4log.

[90] Lardier, C. and Barensky, S., *The Soyuz Launch Vehicle: The Two Lives of an Engineering Triumph.* New York: Springer, 2013.

[91] Larson, W., Kirkpatrick, D., Ryan, R., and Weyers, V., *Space Launch and Transportation Systems*, vol. 1. Department of Defense, January 2005.

[92] Larson, W., Kirkpatrick, D., Ryan, R., and Weyers, V., *Space Launch and Transportation Systems*, vol. 2. Department of Defense, January 2005.

[93] Launius, R. and Jenkins, D., *To Reach the High Frontier: A History of U.S. Launch Vehicles.* Lexington, KY: The University Press of Kentucky, 2002.

[94] Lee, J.-R. and Dhital, D., “Review of flaws and damages in space launch vehicle: Structures,” *Journal of Intelligent Material Systems and Structures*, vol. 24, no. 1, pp. 4 – 20, 2012.

[95] Lewis, E., *Introduction to Reliability Engineering.* Wiley, 2 ed., November 1995.

[96] Liu, T. and Chiou, S., "The application of petri nets to failure analysis," *Reliability Engineering &amp; System Safety*, vol. 57, pp. 129 – 42, August 1997.

[97] Lowery, H. and Haufler, W., "Independent Orbiter Assessment: Assessment of the Data Processing System FMEA/CIL," tech. rep., McDonnell Douglas Astronautics Company, Houston, TX, November 1986.

[98] Lurie, D., Abramson, L., and Vail, J., *Applying Statistics.* No. NUREG-1475, United States Nuclear Regulatory Commission, March 2011.

[99] Lutomski, M., Farnham, S., and Grant, W., "Estimating the Reliability of a Soyuz Spacecraft Mission," *Proceedings - 10th International Probabilistic Safety Assessment and Management*, June 2010.

[100] Mathias, D. L., Go, S., Gee, K., and Lawrence, S., "Simulation assisted risk assessment applied to launch vehicle conceptual design," *Proceedings - Annual Reliability and Maintainability Symposium*, 2008.

[101] Mattenberger, C., Leszczynski, J., Putney, B., and Morse, E. L., "Launch vehicle reliability growth," *Proceedings - Annual Reliability and Maintainability Symposium*, 2012.

[102] May, T. and Creech, S., "NASA's Space Launch System (SLS) Program: Mars Program Utilization," *Concepts and Approaches for Mars Exploration Workshop*, June 2012.

[103] McFadden, R. and Shen, Y., "An Analysis of The Historical Reliability of US Liquid-Fuel Propulsion Systems," *AIAA/SAE/ASME 26th Joint Propulsion Conference*, July 1990.

[104] McMahan, T., "NASA Engineers Resurrect and Test Mighty F-1 Engine Gas Generator: Testing will aid NASA's Space Launch System Advanced Development." Online, January 2013.

[105] Memo CB-04-044, "From: CB/chief, astronaut office, to: CA/director, Flight Crew Operations," May 2004.

[106] Misra, K., *Reliability Analysis and Prediction.* Amsterdam, The Netherlands: Elsevier, 1992.

[107] Moore, D., Phelps, J., Kanner, H., Freeland, D., and Olson, D., "The Reusable Solid Rocket Booster (RSRB) - A Booster System," *AIAA Space Conference & Exposition*, September 2011.

[108] Morse, E. L., Fragola, J. R., and Putney, B., "Modeling launch vehicle reliability growth as defect elimination," *AIAA SPACE 2010 Conference & Exposition*, August 2010.

[109] NASA Office of Safety and Mission Assurance, Washington, D.C., *NASA Risk-Informed Decision Making Handbook*, 1 ed., April 2010. NASA/SP-2010-576.

[110] National Aeronautics and Space Administration, "J-2 Engine Fact Sheet," December 1968.

[111] National Aeronautics and Space Administration, "Report of the SSME Assessment Team," tech. rep., Washington, D.C., January 1993.

[112] National Aeronautics and Space Administration, NASA-STD 8729.1, *Planning, Developing, and Managing an Effective Reliability and Maintainability (R&M) Program*, December 1998.

[113] National Aeronautics and Space Administration, "X-34: Demonstrating reusable launch vehicle technologies." Online, October 1999.

[114] National Aeronautics and Space Administration, "Columbia accident investigation board final report," tech. rep., Washington, D.C., August 2003.

[115] National Aeronautics and Space Administration, "Space shuttle: Shuttle basics." Online, March 2006.

[116] National Aeronautics and Space Administration, Washington, D.C., *NASA Systems Engineering Handbook*, December 2007.

[117] National Aeronautics and Space Administration, Washington, D.C., *Agency Risk Management Procedural Requirements*, NPR 8000.4A, 2008.

[118] National Aeronautics and Space Administration, "Falcon 9 launch vehicle nafcom cost estimates," August 2011.

[119] National Aeronautics and Space Administration, "NASA facts: J-2X Engine," November 2011.

[120] National Aeronautics and Space Administration, Washington, D.C., *NASA Risk Management Handbook*, 1.0 ed., November 2011.

[121] National Aeronautics and Space Administration, Washington, D.C., *Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners*, 2nd ed., December 2011.

[122] National Aeronautics and Space Administration, "Progress 43 launch." International Space Station Imagery, June 2011.

[123] National Aeronautics and Space Administration, "Space Shuttle Discovery." Space Shuttle Image Gallery, June 2011. http://www.nasa.gov/mission_pages/shuttle/multimedia/highlights_gallery.html.

[124] National Aeronautics and Space Administration, "NASA facts: Space Launch System," June 2012.

[125] Oberg, J., "Sputnik witnesses saw failure, then success." NBC News Online, October 2007.

[126] Orbital Sciences Corporation, *Antares OSP-3 User's Guide*, July 2013. Release 1.1.

[127] Orloff, R., *Apollo by the Numbers: A Statistical Reference*. No. SP-2000-4029, Washington, DC: NASA History Division, October 2000.

[128] Parkinson, R., "The hidden cost of reliability and failure in launch systems," *Acta Astronautica*, vol. 44, no. 7-12, pp. 419–424, 1999.

[129] Pate-Cornell, E. and Dillon, R., "Probabilistic risk analysis for the nasa space shuttle: A brief history and current work," *Reliability Engineering and System Safety*, vol. 74, no. 3, pp. 345 – 352, 2001.

[130] Peterson, J. L., *Petri Net Theory and the Modelling of Systems*. Englewood Cliffs, New Jersey: Prentice-Hall, April 1981.

[131] Peterson, T., "Development History of the Space Shuttle Main Engine," *22nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, June 1986.

[132] Pratt & Whitney Aircraft, "Design Study of RL10 Derivatives: Engine Design Characteristics Final Report Volume II," tech. rep., National Aeronautics and Space Administration, Huntsville, AL, December 1973.

[133] Pratt & Whitney Rocketdyne, “Space shuttle main engine fact sheet.” Online, 2012.

[134] Pugh, R., “The Many Facets of the RL10 Liquid Rocket Engine: A Continuing Success Story,” *34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, 1998.

[135] Pugh, R., “Space transportation engine reliability prediction methodology,” *AIAA/SAE/ASME 27th Joint Propulsion Conference*, June 1991.

[136] ReliaSoft, “Synthesis Reliability Software.” Free trial version available at: http://www.reliasoft.com/synthesis/demo.htm.

[137] Rocketdyne Corporation, “Aerojet Rocketdyne Completes J-2X Testing.” Online, June 2014. http://www.rocket.com/article/aerojet-rocketdyne-completes-j-2x-testing.

[138] Rome Laboratory, *Reliability Engineer’s Toolkit.* Griffiss AFB, NY: Air Force Materiel Command, April 1993.

[139] Ryan, R. S. and Townsend, J. S., “Fundamentals and issues in launch vehicle design,” *Journal of Spacecraft and Rockets*, vol. 34, no. 2, pp. 192 – 198, 1997.

[140] Sackheim, R. L., Ryan, R., and Threet, E., “Survey of Advanced Booster Options for Potential Shuttle-Derivative Vehicles,” *37th AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, July 2001.

[141] Sackheim, R. L., “Overview of united states space propulsion technology and associated space transportation systems,” *Journal of Propulsion and Power*, vol. 22, no. 6, pp. 1310 – 1333, 2006.

[142] Safie, F. M. and Weldon, D. M., “Design for reliability and safety approach for the new nasa launch vehicle,” tech. rep., NASA Marshall Space Flight Center, Huntsville, Alabama, 2009.

[143] Santiago, J., “Evolution of the RL10 Liquid Rocket Engine for a New Upper Stage Application,” *32nd AIAA/ASME/SAE/ASEE Joint Propulsion Conference and Exhibit*, July 1996.

[144] Sawyer, J., “STS-1 Chronology.” NASA History Program Office. http://history.nasa.gov/sts1/chronology.html.

[145] Schmeckpeper, K., “Independent Orbiter Assessment: Assessment of the Electrical Power Distribution and Control Subsystem,” tech. rep., McDonnell Douglas Astronautics Company, Houston, TX, February 1988.

[146] Schrader, A., “Lockheed martin preparing to build new gps satellites in denver-area facility,” *The Denver Post*, December 2011.

[147] Schrage, D., “Technology for Affordability - How to Define, Measure, Evaluate, and Implement It,” *50th National Forum of the American Helicopter Society*, May 1994.

[148] Schrage, D., “Technology for Rotorcraft Affordability Through Integrated Product/Process Development (IPPD),” *55th National Forum of the American Helicopter Society*, May 1999.

[149] Selby, J. and Miller, S., “Reliability Planning and Management (RPM),” *Proceedings ASQC/SRE Seminar*, Milwaukee, WI 1970.

[150] Shao, J., *Mathematical Statistics*. New York, New York: Springer, 2003.

[151] Singer, J., “Space Launch System (SLS) Program Overview,” *Advanced Development NRA Industry and Academia Day*, February 2012.

[152] Stamatis, D., *Failure Mode and Effect Analysis: FMEA from Theory to Execution.* Milwaukee, Wisconsin: ASQC Quality Press, 1995.

[153] Stanley, D., Cook, S., Connolly, J., Hamaker, J., Ivins, M., Peterson, W., Geffre, J., Cirillo, B., and McClesky, C., "Nasa's exploration systems architecture study," tech. rep., National Aeronautics and Space Administration, November 2005.

[154] Sutton, G., "History of Liquid Propellant Rocket Engines in the United States," *Journal of Propulsion and Power*, vol. 19, pp. 978–1007, November 2003.

[155] Sutton, G., *History of Liquid Propellant Rocket Engines.* Library of Flight, AIAA, November 2005.

[156] Sutton, G. and Biblarz, O., *Rocket Propulsion Elements.* Hoboken, NJ: Wiley, 8th ed., 2010.

[157] Svitak, A., "ILS sets september return to flight for proton." Aviation Week Online, August 2013.

[158] Trahan, W., O'Donnell, R., Pietz, K., and Drapela, L., "Independent Orbiter Assessment: Assessment of the Guidance, Navigation, and Control Subsystem FMEA/CIL," tech. rep., McDonnell Douglas Astronautics Company, Houston, TX, January 1988.

[159] Ullah, R., Zhou, D.-Q., Zhou, P., Hussain, M., and Amjad Sohail, M., "An approach for space launch vehicle conceptual design and multi-attribute evaluation," *Aerospace Science and Technology*, 2011.

[160] United Launch Alliance, *Atlas V Launch Services User's Guide.* Centennial, CO, March 2010.

[161] United Launch Alliance, *Delta IV Launch Services User's Guide*. Centennial, CO, June 2013.

[162] United States Air Force, "RAPTOR Reliability Software Version 4.0S." Available for download via Barringer & Associates: http://www.barringer1.com/raptor.htm.

[163] United States Air Force, *Early Systems Engineering Guidebook*, 1 ed., March 2009.

[164] U.S. Army Materiel Systems Analysis Activity, "Reliability growth planning models." Online, September 2013.

[165] Van Hooser, K. and Bradley, D., "Space Shuttle Main Engine - The Relentless Pursuit of Improvement," *AIAA Space Conference & Exposition*, September 2011.

[166] Vartabedian, R., "Cost of columbia accident inquiry is soaring," *Los Angeles Times*, March 2003.

[167] Vesely, W., *Fault Tree Handbook*. Washington, D.C.: U.S. Nuclear Regulatory Commission, January 1981.

[168] Villeneuve, F., *A Method for Concept and Technology Exploration of Aerospace Architectures*. PhD thesis, Georgia Institute of Technology, Atlanta, Georgia, August 2007.

[169] Volovoi, V., "Modeling of system reliability petri nets with aging tokens," *Reliability Engineering & System Safety*, vol. 84, pp. 149 – 61, May 2004.

[170] Volovoi, V., "Stochastic petri net modeling using spn@." 2005.

[171] Volovoi, V., "Reliability modeling lecture set 2," in *Safety by Design and Flight Certification Course*, Georgia Institute of Technology, February 2013.

[172] Wilhelmsen, C. and Ostrom, L., *Risk Assessment: Tools, Techniques, and Their Applications*. Somerset, NJ: Wiley, June 2012.

[173] Williamson, R., *Exploring the Unknown*, vol. 4, ch. 2 ”Developing the Space Shuttle”, pp. 161 – 193. National Aeronautics and Space Administration, 1999.

[174] Wood, B., “RS-68: What and How,” *34th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, 1998.

[175] Wood, B., “Propulsion for the 21st Century - RS-68,” *38th AIAA/ASME/SAE/ASEE Joint Propulsion Conference*, July 2002.

[176] Yang, G., *Life Cycle Reliability Engineering*. Hoboken, New Jersey: Wiley, 2007.

[177] Zegler, F., “An Integrated Vehicle Propulsion and Power System for Long Duration Cryogenic Spaceflight,” *AIAA SPACE 2011 Conference & Exposition*, September 2011.

[178] Zegler, F., “Integrated Vehicle Fluids,” *United States Patent no. US 2012/0227374 A1*, March 2012.

[179] Zwicky, F., *The Morphological Method of Analysis and Construction*. Interscience Publishing, 1948.