3-23-2017

Reliability and Cost Impacts for Attritable Systems

Bryan R. Bentz

RELIABILITY AND COST IMPACTS FOR ATTRITABLE SYSTEMS

THESIS

Bryan R. Bentz, 1st Lieutenant, USAF

AFIT-ENV-MS-17-M-172

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENV-MS-17-M-172

RELIABILITY AND COST IMPACTS FOR ATTRITABLE SYSTEMS

THESIS

Presented to the Faculty

Department of Systems Engineering and Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Systems Engineering

Bryan R. Bentz, BS

1st Lieutenant, USAF

March 2017

AFIT-ENV-MS-17-M-172

RELIABILITY AND COST IMPACTS FOR ATTRITABLE SYSTEMS

Bryan R. Bentz, BS

1st Lieutenant, USAF

Committee Membership:

John M. Colombi, Ph.D.
Chair

Jason K. Freels, Maj, USAF, Ph.D.
Member

Jason W. Sutherlin, Capt, USAF (Member)
Member

Acknowledgments

Many thanks to my faculty advisor, Dr. Colombi, for his willingness to support my research of this rich area of study. Without his guidance and candor, I would not have been able to complete this thesis effort. Thanks to my sponsor at AFRL/RQVI for treating the academic support as valuable members of the team. Additionally, thanks are owed to my contact the engineers at the Subscale Aerial Targets branch, without whom my quest for "attritable" field data would have come to naught. And finally, to my wife – without your supreme patience I may not have finished at all.

Bryan R. Bentz

Table of Contents

List of Figures

Page

List of Tables

List of Acronyms

ACTD	Advanced Concept Technology Demonstrator
AFRL	Air Force Research Laboratory
AGREE	Advisory Group on the Reliability of Electronics Engineers
AUFC	Average Unit Flyaway Cost
CAPE	Cost Assessment and Program Evaluation
CER	Cost Estimation Relationship
CIF	Cumulative Incidence Function
DAU	Defense Acquisition University
DoD	Department of Defense
DoDI	Department of Defense Instructive
DOT&E	Defense Operational Test and Evaluation
FTA	Fault Tree Analysis
HAF	Headquarters, Air Force
ISR	Intelligence, Surveillance, and Reconaissance
JPO	Joint Program Office
KM	Kaplan-Meier Estimator
LCAAT	Low Cost Attritable Aircraft Technology
MALE	Medium Altitude, Long Endurance
MCF	Mean Cumulative Function
MCMC	Markov Chain Monte Carlo
MFOP	Maintenance Free Operating Period
MTBF	Mean Time between Failures
MTTF	Mean Time to Failure
NDI	Non-Developmental Item
O&S	Operations and Support
OSD	Office of the Secretary of Defense
R&D	Research and Development
RAC	Reliability Analysis Center
RBD	Reliability Block Diagram
ROCOF	Rate of Occurrence of Failure
RPA	Remotely Piloted Aircraft
RQKP	Office Symbol of the Air Vehicle Directorate's contracting branch
SAN	Stochastic Activity Network
SEBoK	Systems Engineering Body of Knowledge
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
UML	Unified Modeling Language

AFIT/GSE/ENV/17--172

Abstract

Attritable systems trade system attributes like reliability and reparability to achieve lower acquisition cost and decrease cost risk. Ultimately, it is hoped that by trading these attributes the amount of systems able to be acquired will be increased. However, the effect of trading these attributes on system-level reliability and cost risk is difficult to express complicated reparable systems like an air vehicle. Failure-time and cost data from a baseline limited-life air vehicle is analyzed for this reliability and reparability trade study. The appropriateness of various reliability and cost estimation techniques are examined for these data. This research employs the cumulative incidence function as an input to discrete time non-homogeneous Markov chain models to overcome the hurdles of representing the failure-time data of a reparable system with competing failure modes that vary with time. This research quantifies the probability of system survival to a given sortie, $S(n)$, average unit flyaway cost (AUFC), and cost risk metrics to convey the value of reliability and reparability trades. Investigation of the benefit of trading system reparability shows a marked increase in cost risk. Yet, trades in subsystem reliability calculate the required decrease in subsystem cost required to make such a trade advantageous. This research results in a trade-space analysis tool that can be used to guide the development of future attritable air vehicles.

1.0 Reliability and Cost Impacts for an Attritable System

1.1 Chapter Overview

This introductory chapter establishes a definition for the term “attritable” founded on recent research activities within the context of a limited-life air vehicle and examines the system attributes of design life and reliability. These definitions are used to formulate the fundamental problem statement and the related investigate questions that must be answered within this research. The methodology by which these investigative questions will be addressed is outlined, yet covered in detail in proceeding chapters. Finally, the simplifying assumptions that this reliability and cost research is predicated upon are examined. The limitations that data availability and the choice of modeling technique impose on the research outcomes are also discussed.

1.2 Background

In 2014, the Secretary of Defense at the time introduced the “third offset strategy” to a group of defense experts at the Reagan Defense Forum. The third offset seeks to build off of the strategies emphasized in the first and second offsets of the 1950s and mid-1970s, respectively to ensure the United States could overcome a looming quantitative disadvantage in future conflicts. The third offset strategy’s intent is to build off of earlier accomplishments in nuclear deterrence, intelligence, surveillance and reconnaissance (ISR), precision weapons, stealth, and space-based military communication and navigation through improving the performance and decision-making ability of the warfighter in highly-contested environments.

Deputy Secretary of Defense Bob Work (2014 – Present) identified the five pillars of the Third Offset strategy, which he highlighted will be "looked favorably upon in the budget," when he addressed the Center for New American Security in 2015. The five pillars include: support of technologies to improve learning systems, human-machine collaboration, combat teaming, assisted human operations, and cyber-hardened yet network-enabled autonomous weapons (Mehta, 2015). Yet, Work specifically challenged developers of autonomous aircraft in his address. Work ventured, "What we want to do on human-machine combat teaming is to take it to the next level, to look at things like swarming tactics. Can an F-35 go into battle with four unmanned wingmen?" (Mehta, 2015).

Air Force guidelines, such as Headquarters, Air, Force's (HAF) USAF Future Operating Concept and the Remotely Piloted Aircraft (RPA) Vector Vision Enabling Concepts 2013-2038 align with the course plotted by DoD leadership. In this guideline it is acknowledged that the next generation of autonomous aircraft that "must detect, avoid, or counter known threats via traditional or innovative means, to enable operations in a range of environments" (HAF, 2014, p. 32) HAF states that, "this can be achieved through a combination of speed, low observable technology, altitude, maneuverability, employment of air-launched Small Unmanned Aerial Systems, active and passive countermeasures, or expendable assets." (HAF, 2014, p.32).

Perhaps the most novel and yet least understood method to operate in highly-contested environments and overcome a quantitative disadvantage is employment of expendable assets. In 2016, Dr. David Walker, Deputy Assistant Secretary of the Air

Force for Science, Technology, and Engineering, briefed the House Armed Services Committee on how "disaggregated unmanned air systems present a new dimension for achieving the operational agility envisioned in the *Air Force Future Operating Concept* and the Department's Third Offset Strategy" (Walker, 2016, p. 8). During his presentation, Dr. Walker specifically highlighted the Low Cost Attritable Aircraft Technology (LCAAT) demonstration program as an emerging capability with great opportunity. The LCAAT vision system, according to Dr. Walker, is meant to leverage recent technological advances in manufacturing to field near-term expendable or limited-life unmanned air platforms as single assets or manned/unmanned teams (Walker, 2016, p.8). The goal of the LCAAT program is "to trade the relatively high costs of UAV performance, design life, reliability, and maintainability for a low cost attritable aircraft intended for re-use with limited life and number of sorties" (Keller, 2015).

The researchers determined that a relatively low lifecycle cost is achieved by focusing on the typical lifecycle cost drivers. The Office of the Secretary of Defense's (OSD) Director of Cost Assessment and Program Evaluation (CAPE) cost estimating guide illustrates which of the four lifecycle cost categories – Research and Development (R&D), Investment, O&S, and Disposal costs are key drivers of overall lifecycle cost. Figure 1 depicts an expenditure profile for a typical DoD program over the course of its lifecycle.

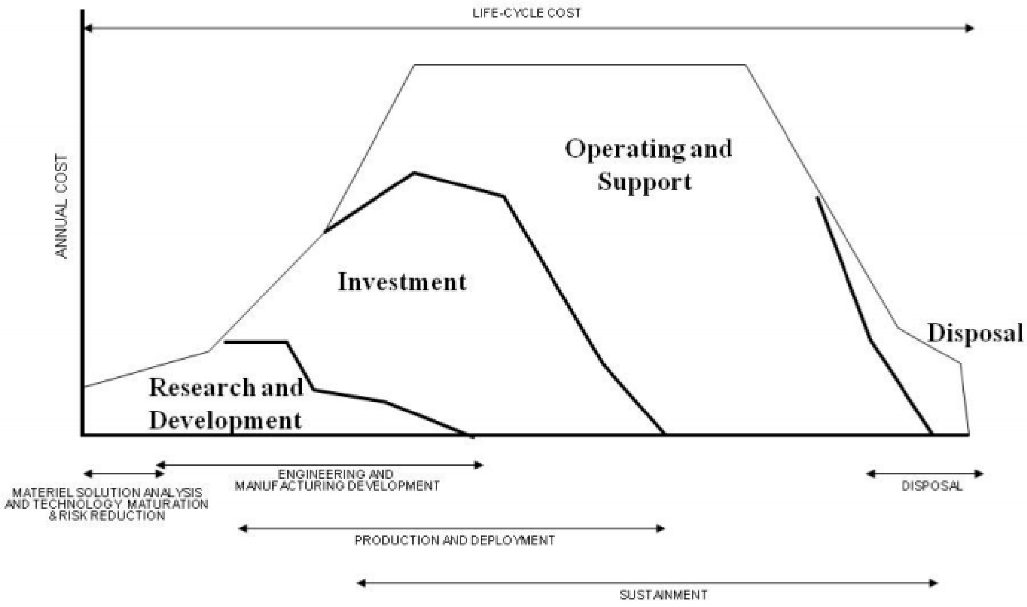


Figure 1: Illustrative System Lifecycle (Source: CAPE, 2014, 2-1)

The profiles of Figure 1 clearly show that the two cost categories that have the greatest effect on a typical DoD system’s lifecycle cost are the Investment, and Operating and Sustainment (O&S) costs.

While the majority of a system’s lifecycle cost is incurred during the Investment and O&S phases, it is easiest to control these costs earlier in the system lifecycle. According to pioneering systems engineers, Benjamin S. Blanchard and Wolter J. Fabrycky, the ease with which systems engineers can change the attributes that contribute significantly to Investment and O&S costs decreases exponentially as the system proceeds through the lifecycle. Figure 2 illustrates Blanchard and Fabrycky’s argument that the opportunity to easily change a system to control lifecycle cost opens early and decreases as the system configuration is finalized.

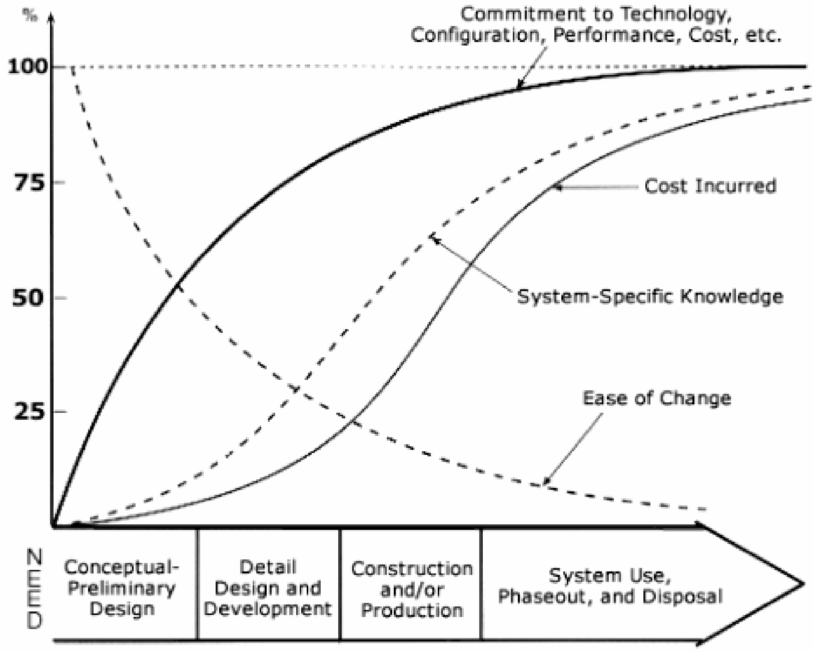


Figure 2: Commitment, system-specific knowledge and cost (Source: Blanchard & Fabrycky, 1998, Figure 2.11)

Blanchard and Fabrycky's proposal that the most favorable time to control Investment and O&S costs is early in the lifecycle is supported by acquisition policy. "DoDI 5000.02 requires that sustainment factors be fully considered at all milestone reviews and other acquisition decision points, and that appropriate measures be taken to reduce O&S costs by influencing system design early in development, developing sound sustainment strategies, and addressing key drivers of cost…the opportunities to reduce O&S costs decline significantly as the system design evolves" (CAPE, 2014, 2-1).

The importance of identifying and addressing lifecycle cost drivers for the LCAAT program cannot be overstated as the researchers themselves define attritability as a system trait "whereby virture of its cost, loss of the aircraft could be tolerated" (AFRL/RQKP, 2015, p. 1). This definition for "attritable" is consistent with early uses of the term by the Director, Operational Test and Evaluation (DOT&E, 1999), the history of which is addressed in subsequent chapters. The intent is to produce a relatively low cost system compared against the high cost to counter its capabilities and “force a cost imposition of near peer adversaries” (AFRL/RQKP, 2015, p.1). In order to accomplish this, “performance, design life, reliability, and maintainability with their associated costs need to be traded” (AFRL/RQKP, 2015, p. 1). While there exist numerous strategies to engineer and measure reliability and maintainability, little is known about they apply to an attritable system.

1.3 Definitions

There currently exists little guidance on what constitutes an *attritable* system within the DoD. Additionally, the use of terms such as *design life* and *reliability* should be considered carefully as trading these attributes are critical to the achievement of attritability for a system. Further complicating things is the fact that the definition of these terms have evolved over time due to interchangeable and imprecise use. Therefore, a definition for reliability and design life is provided to serve as a foundation upon which to base the trade study methodology and subsequent conclusions. A more comprehensive discussion of reliability-related terms and definitions is provided in a subsequent chapter.

This research seeks to determine the reliability impacts on attritable systems and the resultant effect trading reliability on cost. Therefore, it is important to identify the unique characteristics of an attritable system. While alternative definitions for the term and a background of the attritable air vehicle concept are offered in later chapters, this research uses the definition provided by AFRL. According to AFRL, an attritable system is one that is intended to be used more than once – which separates the system from expendable systems – and, “whereby virtue of its cost, the loss of the aircraft could be tolerated” (AFRL/RQKP, 2015, p.1). Compared to alternative definitions for “attritable,” this definition simplifies the scope of trade space analyses to only the measurable system costs and reliability.

Finally, it is necessary to highlight the differences between the attributes of system reliability and design life. The semantic shift of these terms are examined in subsequent chapters, however this research relies on the seminal report from the Advisory Group on the Reliability of Electronic Engineers (Hogge, 2012, p.8). “A 1957 AGREE report defined reliability as the probability that a system or product will perform in a satisfactory manner for a given period of time when used under specified operating conditions in a given environment” (Hogge, 2012, p. 8). The four key elements of this definition are: (1) reliability as a probability distribution, (2) specified satisfactory performance, (3) specified operation conditions, and (4) in a specific environment (Hogge, 2012, p.8). The AGREE definition of reliability offers the most flexibility in specifying the failure mode when compared to definition provided for the design life attribute by the System Engineering Book of Knowledge’s (SEBoK). The primary difference between design life and reliability is that reliability describes the probability of failure, for any reason, and not just due to a wear-out mechanism as the SEBoK uses design life. There are numerous metrics to describe the reliability of many types of systems; a discussion of these metrics is provided in the proceeding chapter.

1.4 Research Objectives

It is difficult to express the impact of subsystem reliability and reparability decisions on an attritable system's overall reliability or its corresponding expected costs. To overcome this, applicable reliability modeling and expected cost estimation techniques, valid for attritable systems early in the product lifecycle, will be applied. Through characterizing the effects of varying subsystem reliability and reparability on system-level reliability, and the subsequent effect on expected costs, this research can inform decision makers on the value of these design trades.

1.5 Investigative Questions

To quantify the impact of component, design, and maintenance choices on an attritable aircraft's reliability and cost, the following issues must be investigated:

1. What metrics and methods are suitable for to the estimation of reliability and costs for attritable systems?
2. How sensitive is an attritable system's reliability to changes in subsystem reliability and reparability?
3. What effect does varying subsystem reliability and reparability have on the average unit flyaway cost and the cost at risk of an attritable system?

1.6 Methodology Overview

1. Lifetime data of a similar fielded system is analyzed. In this analysis, satisfactory system performance is based upon the requirements of the LCAAT program. The specific failure modes are identified and are allocated to various subsystems. To estimate the reliability of the constituent subsystems the competing risk of failure

is considered to more accurate estimate each subsystem's respective hazard rate. These data are applied to the subsequent reliability model and used as a baseline to compare the effects of varying subsystem reliability and reparability against.

2. The context of the available data on existing limited-life air vehicles is considered to determine the method which allows for the variation of subsystem reliability and reparability, and whose output can be applied to the estimation of expected costs. A reliability model is constructed – based off of an existing limited-life air vehicle system architecture – in the form of a discrete time-nonhomogeneous Markov Chain model. This modeling technique is the most appropriate as it allows for the specification and variation of reliability and reparability parameters. Variation of these transitional probabilities allow the model to examine the stated research goals of this research.
3. The baseline discrete time-nonhomogeneous Markov Chain model is altered by varying each respective subsystem's competing risk hazard rate, as well as varying each subsystem's reparability – that is, its ability to be restored to an operational state. This allows for the determination of the sensitivity of system-level reliability, defined in termed of its survival function $S(t)$, to these changes.
4. Example system regeneration, system replacement (defined as average unit flyaway cost), and subsystem repair costs are defined and applied to transitional probabilities output by discrete time-nonhomogeneous Markov Chain model to predict the effect of trading reliability and reparability on system cost risk and average unit flyaway cost.

1.7 Assumptions and Limitations

This study seeks to predict the impact of trading subsystem reliability and maintainability of an air vehicle on its overall reliability and expected costs in the hopes of determining the appropriate level of reliability for an attritable system. A central assumption of this research, which makes it applicable to the conceptual design of an attritable air vehicle, is that the vision system will contain subsystems similar to those used by existing limited-life air vehicles. This is a reasonable assumption as the current design philosophy of the LCAAT program emphasizes the use of non-developmental items (NDI).

An assessment of the physical architecture of existing limited-life remotely operated air vehicles show that the system is composed of seven subsystems: the system operator, launch system, vehicle structure, electrical system (which consists of everything from communications equipment to avionics and control surface actuators), fuel management system, propulsion system, and recovery subsystem. The assumption that the vision system will be composed of these subsystems is not only supported by the program's emphasis on NDI, but the consistency of these subsystems with those outlined by MIL-STD 881C Work Breakdown Structures for Defense Materiel Items, Appendix H Unmanned Aerial Vehicle Systems.

Analysis of the life data gathered on the baseline fielded system shows that the occurrence of a subsystem failure can prevent the observation of other subsystem failures. Consequently, the competitive nature of these failure modes must be accounted for in the computation of subsystem reliability. Therefore, subsystem reliability will be defined in terms of the hazard rate of each respective failure mode's cumulative incidence function. This is an assumption critical to the specification of the discrete time-nonhomogeneous Markov Chain model. The theory and arithmetic of these competing risks analyses is based on literature discussed in proceeding chapters.

This research is limited to studying the impact of varying subsystem reliability and reparability. The effect of varying component redundancy cannot be studied as the life data gathered on similar fielded systems is at a higher level of abstraction. That is, the data upon which the reliability models are based on are not detailed enough to allocate each failure occurrence to its root component – merely the respective subsystem. If more detailed life data that identified the root cause of each failure at the component level were used, a more detailed analysis could use the same methodology outlined here.

Lastly, this research's intent to study reliability, as well as reparability – that is the ability of a failed subsystem to be restored to an operational state – highlights the need for the modeling technique to allow for variations in reparability. Furthermore, the failure-time data gathered on the baseline system not only demonstrates the existence of competing failure modes, but also the existence of time dependent failure modes – where the probabilities of subsystem failure change with respect to time. This analysis is presented and underscores the need for the modeling technique to have the ability to model competitive failure modes that chance over time. Therefore, this research is predicated off of the use of a discrete time-nonhomogeneous Markov Chain model as it satisfies the aforementioned requirements. The alternatives to this technique, as well as the consequences of selecting this reliability modeling technique are discussed in subsequent chapters.

1.8 Research Preview

The proceeding chapters will discuss the relevant literature regarding reliability metrics, modeling techniques, and methods to compute the impact of changing reliability and reparability parameters on expected costs. Also discussed is the applicability of these methods to attritable systems. A methodology to create a reliability model for an attritable air vehicle is proposed, which is founded on life data gathered for existing limited-life air vehicles. The reliability model is populated by the statistics computed by analyzing the life data gathered on existing limited-life air vehicles and research excursions are performed to model the impacts of varying subsystem reliability and reparability on overall reliability and cost risk. The conclusions of these research excursions can be used by decision makers to guide the development of attritable systems.

2.0 Review of Literature for Relating to Attritability, Reliability and Cost

2.1 Chapter Overview

This chapter not only examines the variety of reliability metrics and modelling techniques, but also explores the history of unmanned aerial vehicles (UAVs) used as "attritable" or limited-life systems. By identifying the variety of techniques for the prediction and measurement of reliability based on such factors as purpose, maturity, and data availability this chapter identifies techniques that are most applicable to recent research regarding attritable air vehicles. The methods for the analysis of competing failure modes is examined in order to apply these techniques to this research's reliability models. Finally, literature of expected cost estimation is examined. Context is gleaned on the cost estimation of systems early in the design phase, while methods to forecast the impact of varying reliability and reparability parameters are also examined.

2.2 The Characteristic of Attritability

The term attritable to describe the acceptability of a system's loss by the user has only just begun to enter the DoD systems engineering lexicon based on recent actions taken by the Air Force Research Lab among others. However, the term dates back almost two decades, while the concept of creating systems where the user is tolerant of it loss dates back even further. The first use of the term "attritable" in official DoD program documentation is found in a 1999 report on the Predator system published by the Director, Operational Test and Evaluation (DOT&E). Through reporting the current state of operational testing, the DOT&E described the Predator system as a "system that operates autonomously, is attritable (air vehicle cost is less than $3.5M), and does not compromise sensitive technology should it be lost over enemy territory" (DOT&E, 1999). This report suggests that a system's "attritability" is a function of the tolerance of its loss. An attritable system is one which is designed such that the user is not unduly averse to the system's loss. In the case of this report, it is based on the system's replacement cost and chance of compromising sensitive technology. While the term may date back to this specific usage, the concept of employing a system whose loss the user can tolerate it nothing new.

The use of unmanned air systems for dull, dirty, or dangerous missions underscores how a user can be more tolerant of their loss and their use "trace their modern origins back to the development of aerial torpedoes almost 95 years ago," (Keane & Carr, 2013, p. 558). Yet the supporting technologies that allowed for the achievement of militarily useful capabilities took time to mature. In 1960, "the era of remotely piloted vehicles was born under the code name 'Red Wagon,' when the United States Air Force (USAF) awarded a modest $200,000, but highly classified contract to Ryan Aeronautical Company for a flight test demonstration showing how its target drones could be adapted for unmanned, remotely guided photographic surveillance missions" (Schemmer, 1982, p. ii.). Designed to operate in the highly-contested airspace over Southeast Asia, Ryan's concept was carried aloft under the wing of a larger manned aircraft and deployed a parachute for recovery. By leveraging the existing airframe of a BQM-34A target drone, its design was simple and cost effective. Powered by comparatively less reliable, but less costly, engines Ryan Aeronautical Company pioneered the attritable design philosophy.

Development efforts for limited-life unmanned air vehicles languished during the late 1970’s and early 1980’s. However, interest resumed in the mid-1980’s due to “stunning technical advances in micro-electronics, jam-resistant and secure data links, miniaturized sensors, and all-weather guidance, control and recovery systems [that] removed the major technical barriers to exploiting unmanned vehicles for what otherwise would be high-risk missions by piloted aircraft” (Schemmer, 1982, p. iii). The number of development programs grew so rapidly that Congressional analysts perceived duplication of efforts with unclear program objectives (Thirtle, Johnson, & Birkler, 1997).

Congress halted efforts until they were consolidated into a single Joint Program Office (JPO) in 1987, and the Defense Advanced Research Projects Agency (DARPA) began an Advanced Concept Technology Demonstrator (ACTD) effort known as Predator in 1994. The vehicle, based off of an existing airframe provided a solution to the Medium Altitude Long Endurance (MALE) need identified by the UAV JPO (Thirtle, Johnson, & Birkler, 1997). The ACTD effort proved so successful that to this day the Predator is cited a primary example for transitioning a technology demonstrator into the formal acquisition process (Thirtle, Johnson, & Birkler, 1997), and is the first program to be described by the term “attritable.” The relationship between the attribute of attritability and a system’s reliability, as well as metrics, methods, and models of reliability are examined in the proceeding section.

2.3 System Reliability Engineering

Economist, and former Secretary of Defense, James R. Schlesinger once said, “reliability is, after all engineering in its most practical form” (O’Connor, 2002).

However, the defense industry considers reliability engineering a subset of systems engineering and its processes are often mistaken as safety engineering. These specialties frequently use the same prediction techniques to meet their common primary goal of preventing system failure. However, Barnard differentiates between reliability engineering and safety engineering by implying that reliability focuses on the costs of failure, while safety engineering concentrates on the danger of failure. He states, "reliability engineering focuses on costs of failure cause by system downtime, cost of spares, repair equipment, personnel and cost of warranty claims" (Barnard, 2008, 357). Simply put, where safety engineering seeks to prevent system failure as a way to improve human survivability, reliability engineering seeks to prevent system failure to avoid the cost penalties associated with failure.

Barnard's sentiment that reliability engineering emphasizes minimizing the costs of failure is echoed by reliability engineers like Yang. Yang argues that while there are upfront costs to early investment in reliability engineering, it breaks "the design-test-fix loop and thus greatly reduce the time to market and cost. In almost every project reliability investment is returned with substantial savings in design, verification, and production costs" (Yang, 2007, p. 47). The effect that reliability engineering has on total cost is illustrated in Figure 1.

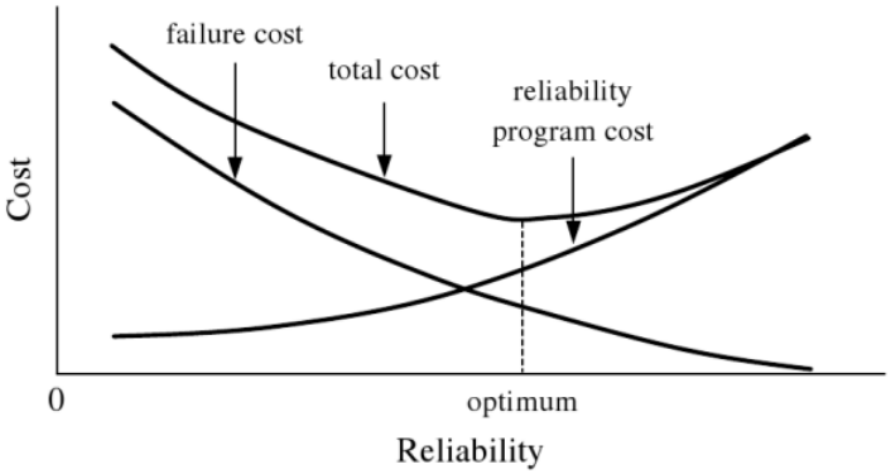


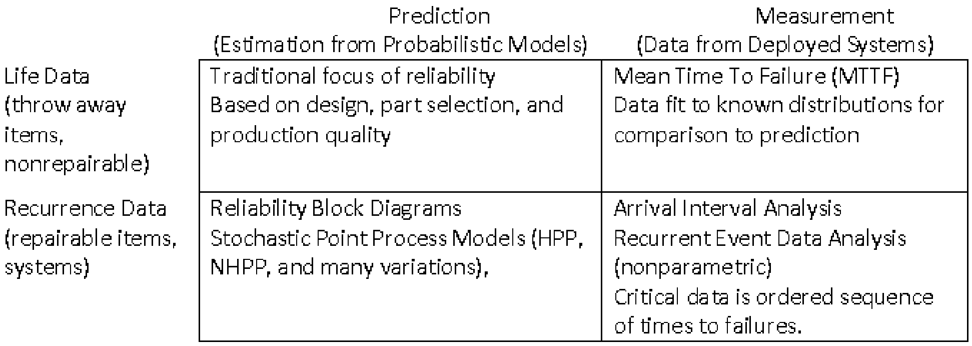
Figure 3: Costs associated with a reactive reliability program (Source: Yang, 2007, p. 47)

Yang’s Figure 1 implies the existence of an optimal level of reliability that minimizes total cost, the definition of which is examined in subsequent sections. This hypothesis of an optimal reliability illustrates the benefit that reliability engineering can have on total cost.

2.3.1 Contexts of System Reliability

Hogge (2012) identifies two distinct applications of reliability engineering; that is, reliability prediction and reliability measurement. Furthermore, Hogge finds that the field can be “divided by the purpose of the analysis and the phase of the product lifecycle” (Hogge, 2012). Hogge’s findings suggest that reliability studies on immature systems can rely on estimation from probabilistic models, yet deployed or mature systems must apply actual time-to-failure data if available.

In addition to maturity and reliability data availability, the intended manner and environment in which the system operates is an additional consideration to determine reliability. Hogge (2012), as well as numerous other reliability engineers, distinguish between items intended for repair and reuse and those considered non-repairable “throw away items.” According to Meeker and Escobar (1998) reliability data for repairable systems is differentiable from non-repairable units because it is based on the sequence of failure times rather than the time to *first* failure (Meeker & Escobar, 1998, p. 3). Hogge illustrates these distinctions in Figure 1 by defining four quadrants or categories of systems. The most applicable reliability measurement or prediction technique for each respective category is identified within quadrant.



	Prediction (Estimation from Probabilistic Models)	Measurement (Data from Deployed Systems)
Life Data (throw away items, nonrepairable)	Traditional focus of reliability Based on design, part selection, and production quality	Mean Time To Failure (MTTF) Data fit to known distributions for comparison to prediction
Recurrence Data (repairable items, systems)	Reliability Block Diagrams Stochastic Point Process Models (HPP, NHPP, and many variations),	Arrival Interval Analysis Recurrent Event Data Analysis (nonparametric) Critical data is ordered sequence of times to failures.

Figure 4: The Four Context Areas of Reliability Analysis (Source: Hogge, 2012, p. 11)

The high cost of modern complex weapon systems typically precludes their treatment as nonrepairable systems, save for expendable systems like munitions. Yet, the statistical techniques associated with nonrepairable systems are often mistakenly applied to the study of repairable systems. Usher (1993) acknowledges this error in *Reliability Models and Misconceptions*, remarking:

> In almost all cases the system under study is repairable in nature; that is, upon failure, it can be restored to operation…A curious point, however, is

> that most of the reliability models found in the literature are appropriate only for the analysis of *non-repairable* systems. (Usher, 1993, p. 261)

Usher determined that the application of non-repairable models and statistical methods to repairable items yielded reliability results that were contrary to correct conclusion.

2.3.2 Reliability Metrics

The limitations and merits of the multitude of reliability metrics are as important to understand as the operating context and environment of the system under study. The manner in which data are measured and recorded determines which metric can be applied. In other cases, a reliability metric may require simplifying assumptions that can hide true failure trends and occlude the underlying relationship between system states and the probability of transition between them. Therefore, a reliability trade study must identify and account for these limitations and assumptions.

According to the MITRE Corporation, "reliability was first practiced in the early start-up days of the National Aeronautics and Space Administration (NASA) when Robert Lusser, working with Dr. Wehner von Braun's rocketry program, developed what is known as "Lusser's Law" (MITRE, 2014). Otherwise known as Lusser's Product Law, it specifies a series system's reliability as the product of the reliability of its components (MITRE, 2014). According to this scheme, a system is considered "weaker that its weakest link", especially if their failure modes are statistically independent.

The impetus for the creation of an advisory group in 1952 was the increasing complexity of DoD weapon systems. The group, known as the Advisory Group on the Reliability of Electronic Equipment (Hogge, 2012, p.8). An AGREE report defined reliability as “the probability that a system or product will perform in a satisfactory manner for a given period of time when used under specified operating conditions in a given environment” (Hogge, 2012, p.8). This served as an acceptable definition until 2003 when Air Force Instruction 21-118, Improving Air and Space Equipment Reliability and Maintainability introduced Mean Time Between Failure (MTBF) as a generalization of reliability. The use of MTBF as a reliability metric was supported by a 2005 Under Secretary of Defense (USD) for Acquisition, Technology, and Logistics (AT&L) memorandum on the matter. It states that, “Material Reliability is generally expressed in terms of mean time between failure(s) (MTBF) and, once operational can be measured by dividing actual operating hours by the number of failures experienced during a specific interval” (USD AT&L, 2005).

The limitations of MTBF and a related metric, used for nonrepairable items, known as Mean Time to Failure (MTTF) presented themselves as the use of these legacy metrics became more prevalent. Kumar (1999) identified the limitations of MTBF and MTTF early in the *Journal of Quality and Maintenance*. According to Kumar, these limitations are twofold: MTBF describes the point in time at which there is equal chance of either survival or failure, and thus fails to also describe the distribution of system failure times. Additionally, Kumar argues, and the metrics assume an exponential distribution of the times-to-failure of the constituent parts of a system.

According to Kumar, the assumption of an exponential time-to-failure distribution restricts the use of MTBF and MTTF in instances of newly designed items or models that must account for wear-out related failures. These limitations led to the abandonment of defense standards such as MIL-HDBK 217 and MIL-STD 1388. Kumar's review of reliability engineering research in *New trends in aircraft reliability and maintenance measures* found that "the search for a new reliability metric is always one of the prime areas of research" (Kumar, 1999).

Kumar qualifies as a reliability statistician himself, yet he posits that such legacy metrics as MTBF and MTTF are imperfect as they are "probabilistic design-based measures…more meaningful to statisticians" than to system operators. (Kumar, 1999b). This disparity between operator and statistician prompted the creation of the Maintenance Free Operating Period metric by Kumar, Knezevic, and Crocker (1999). Defined as "the probability that the item maintains its functionality for at least a period of t_{mf} life units without the need for corrective maintenance due to failure of a component…which results in an overall critical failure of the system" (Kumar, 1999), MFOP is just one example of the myriad of reliability metrics applicable to reliability studies on a reparable system.

Kumar, et al., however, caution that MFOP's usefulness is limited by the simplying assumptions made to make the MFOP calculation tractable. They warn: "if the majority of the failures are non-age related (or so assumed) then there will be very little chance of improving the MFOP probability or the duration of MFOP. In this case, MFOP will not have any advantage compared to that of MTBF" (Kumar, 1999). When it can be assumed that the failure mechanism is neither infant mortality nor wearout, MFOP does not differentiate itself as any more useful than the legacy metrics of MTBF or MTTF; however, the motivation for MFOP still emphasize the design and operational decisions that affect reliability.

A leading systems engineer for Research and Development (R&D) at British Aerospace stated that the challenge of MFOP "lies in the methodology and tools required to produce a system architecture that will reach the target MFOP" (Relf, 1999, 111). Relf identifies design and operational decisions that affect this reliability metric, referred to as "options." Relf's "options", presented in Figure 5, are: (1) inherent reliability, (2) redundancy, (3) reconfigurability, (4) prognostics (the prediction of failure), and diagnostics and (5) lifing policy (an operational decision) (Relf, 1999, 112).

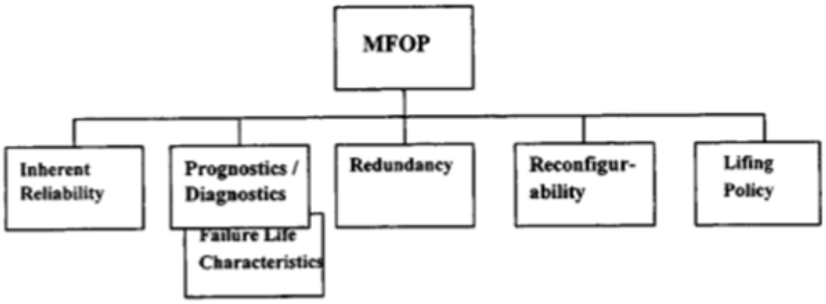


Figure 5: Heirarchy of MFOP Options (Source: Relf, 1999, 112)

Relf warns that a "determination of which MFOP 'option' is the most suitable to maintain functionality" (Relf, 1999, 111) must also be accomplished to achieve a suitable MFOP. Regardless of the usefulness of the MFOP metric, this remark illustrates the importance of trading these options to achieve true attritability, as defined by AFRL.

The inability of MFOP and other reliability metrics to separate themselves from the limitations of legacy metrics like MTBF and MTTF has prompted reliability engineers to use more descriptive parameters. "Typically the traditional parameters of a statistical model (e.g., mean and standard deviation) are *not* of primary interest. Instead, design engineers, reliability engineers, managers, and customers are interested in specific measures of product reliability or particular characteristics of a failure-time distribution (e.g., failure probabilities, quantiles of the life distribution, failure rates)" (Meeker & Escobar, 1998, p. 3). A nonnegative, continuous random variable, typically T, is used to describe the probability of failure at a given time and "can be characterized by a cumulative distribution function, a probability density function, a survival function, or a hazard function" (Meeker & Escobar, 1998, 27).

"The cumulative distribution function (cdf) of T, $\mathrm{F(t)} = \Pr(\ T \leq t)$, gives the probability that a unit will fail before time, t. Alternatively, $\mathrm{F(t)}$ can be interpreted as the proportion of units in the population (or taken from some stationary process) that will fail before time t. " (Meeker & Escobar, 1998, 28). Similarly, the probability density function (pdf) is defined as the cdf's derivative, $\mathrm{f(t)} = \frac{d\mathrm{F(t)}}{d\mathrm{t}}$ and is "used to represent the relative frequency of failure times as a function of time. Although the pdf is less important than the other functions for applications in reliability, it is used extensively in the development of technical results" (Meeker & Escobar, 1998, 28).

Additional valuable functions for reliability analyses are the survival function and the hazard function. The survival function, also called the reliability function. This function is "the complement of the cdf, $\mathrm{S(t)} = \Pr(\ T > t) = 1 - F(t)$, and gives the probability of surviving beyond time t." (Meeker & Escobar, 1998, 28). The hazard function, also known as the hazard rate, is related to both all other reliability distribution functions in the following way: $h(t) = \lim_{\Delta t \to 0} \frac{\Pr(t < T \leq t + \Delta t | T > t)}{\Delta t} = \frac{f(t)}{1-F(t)}$. Thus, the hazard function states the propensity of the unit to fail in the next small interval of time, given that the unit has survived to that time t (Meeker & Escobar, 1998, 28).

The hazard function, also known as the instantaneous failure rate function, is useful "because of its close relationship with failure processes and maintenance strategies" (Meeker & Escobar, 1998, 29). Additionally, the hazard function is the reliability distribution function used as an input for many reliability models. "Where a competitive type of assessment is acceptable, i.e. one design against another the use of general failure rate data is permissible" (Walter & Watson, 1971, 10); however, a common misconception is that failure rate is always constant and is considered the multiplicative inverse of MTBF. The fault with this misconception is that it does not account for all phases of a system lifecycle, as illustrated by what is commonly termed the "bathtub curve"

"The 'bathtub curve' [illustrated by Figure 6] provides a useful conceptual model for the hazard of some product populations" (Meeker & Escobar, 1998, 29).

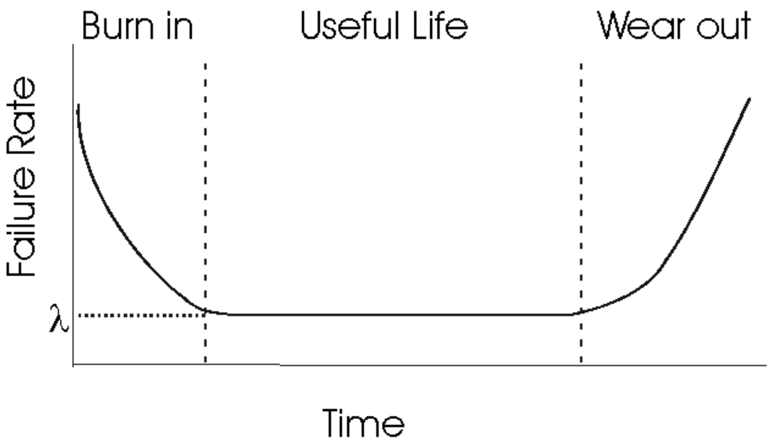


Figure 6: Bathtub Curve for Hardware Reliability (Source: Pan, 1999)

During the burn-in phase, a unit failure may be caused by quality-related defects (infant mortality). The useful life of the product is the period of time during which the hazard rate (or failure rate, λ) is approximately constant and are due to external shocks that occur at random. Failures in the latter stages of the system lifecycle can be attributed to wearout (Meeker & Escobar, 1998, 29).

The preceding reliability metrics are used to describe the time-to-failure behavior of non-repairable units or non-repairable components within a repairable system. Yet, there exists ways to describe the reliability of a repairable system as a sequence of reported system-failure times. Rigdon and Basu's *Statistical Models for the Reliability of Repairable Systems* describe this metric as Mean Cumulative Function (MCF). MCF is defined as the expected value of $N(t)$, $\mu(t) = E[N(t)]$ where $N(t)$, represents the

number of system failures in the interval [0, *t*]. If MCF is differentiable its derivative is defined as $\nu(t) = \frac{d\mu(t)}{dt}$, and is known as Rate of Occurrence of Failure, or recurrence rate per system of the population (Meeker &Escobar, 1998, p. 395). “The MCF and the ROCOF are important elements to define the reliability of a repairable system” (Hogge, 2012, 17).

Regardless of the “options” employed to affect reliability, it is important to begin a reliability study by considering the metric that will apply. The fundamental assumptions of the preceding metrics affect which metrics are applicable to a study that trades reliability for an attritable system. For this reason, it is critical to account for these in the choice of reliability model as well.

2.3.3 Reliability Models

For repairable systems, as well as repairable systems consisting of non-repairable components, probabilistic models are used to predict performance and compare design alternatives. These models are used to predict performance and compare design alternatives. Many reliability models have been developed for the purpose of prediction, and they fit into two categories: combinatorial and state space models (Sahner & Trivedi, 1987). Common combinatorial methods for system reliability prediction are Reliability Block Diagrams (RBDs) and Fault Tree Analyses (FTA), also known as Fault Tree Diagrams. The standard prepared by the International Electrotechnical Commission, Technical Committee on Dependability (known as IEC 61165) outlines these reliability analysis techniques. “FTA can be used to evaluate the probability of a failure at a given instant *t* in time using Boolean logic. This logic may not express time or state dependencies properly” (IEC, 2006, 25). These dependencies may not be expressed due to the fact that “in the fault tree, the parts evaluated have to be assumed to be independent branches” (IEC, 2006, 25).

The assumption of state independence applies to RBDs as well. “A RBD is also a technique that may use Boolean logic and therefore has similar limitations to those of FTA” (IEC, 2006, 25). In the case of RBDs, each block represents a component and the lines between the components describe the relationships of each component to others within the system (often only showing parallel or series structure). Each block possesses a predetermined probability of successful transition. Consequently, it is said that RBDs illustrate the success-oriented states of the system.

As with any technique intended to model and predict the behavior of the system, the assumptions and simplifications made can limit the overall utility of both combinatorial techniques. According to Ascher and Feingold, these methods assume that the behavior of each component is independent of other components and that the probability remain constant. This is referred to as the independent and identically distributed (iid) assumption. Consequently, these techniques are limited to use on systems whose components are not interdependent and do not change over time. Additionally, only recently have extensions to these combinatorial techniques been created with sufficient power to model the ability to restore or repair the system.

There are two generic types of state-space modeling techniques for reliability analyses: Petri nets and Markov techniques. Neither Petri nets nor Markov techniques require the assumption that the components are independent or that the system is static and therefore are not hindered by the limitations of combinatorial methods. “Petri nets are a graphical technique for the representation and analysis of complex logical interactions among elements in a system" (IEC, 2006, 27). In comparison to Markov techniques, General Stochastic Petri nets, a particular class of Petri nets, "can often be described more easily and with a smaller diagram than using Markov techniques" (IEC, 2006, 27). However, "for evaluation purposes, the Petri net is converted to its corresponding Markov model, which is then analyzed" (IEC, 2006, 27).

Studies by researchers like Mura and Chew have demonstrated the effectiveness of Petri nets for modeling dynamic and interdependent systems for reliability analyses. Yet, the statement that the evaluation of a Petri net model begins with its conversion to a corresponding Markov model persists. A complete analysis of reliability evaluation techniques centers on a thorough review of Markov techniques.

Similar to Petri nets, "Markov techniques make use of a state transition diagram which is a representation of the reliability, availability, maintainability or safety behaviors of a system, from which system performance measures can be calculated" (IEC, 2006, 21). Markov techniques are primarily concerned with the definition of system states and the characterization of the transitions between those states. There are a few underlying assumptions of Markov techniques – the most foundational of which is described by Butler and Johnson, researchers at the National Aeronautics and Space Administration (NASA). "A Markov process is a stochastic process whose behavior depends only upon the current state of the system, and not the particular sequence by which the system entered the current state" (Butler & Johnson, 1995). The Reliability Analysis Center (RAC) separates the Markov techniques for reliability analysis into two categories. “There are two basic Markov analysis methods: Markov Chain and Markov process. The Markov chain assumes discrete states and a discrete time parameter; with the Markov Process, states are continuous” (RAC, 2003, 2). Therefore, Markov Chain analyses are most often applied to systems where its elements are described as being in one of two states: working and failed.

Additionally, there are subcategories of Markov Chain that are characterized by how the states are defined, as well as on the behavior of the transition between those states. According to RAC, there exists both homogeneous and non-homogeneous Markov Chains. “A Homogeneous Markov Chain is characterized by constant transition rates between the states. A Non-Homogeneous Markov Chain is characterized by the fact that the transition rates between the states are function of a global clock e.g., elapsed mission time” (RAC, 2003, 2). Transition rates are directly related to the previously defined hazard function and can follow any number of distributions created to describe reliability. The effectiveness of both Homogeneous and Non-Homogeneous Markov Chains to find reliability performance measure of both continuous and phased-mission systems have been proven by researchers such as Dugan, Chew, and Zhou.

These researchers have shown that Petri nets and Markov techniques can be effectively used to represent repairable, and interdependent systems as well as be used to solve for the numerical solution of reliability distribution functions. Yet, the complexity of some system’s state-space representation make the calculation of its reliability distribution functions intractable. For this reason, methods were developed to simulate these complex models. Chew, Dunnett, and Andrews created a Petri net model and “used a form of Monte-Carlo simulation to obtain its results" (Chew, Dunnett, & Andrews, 2007, 219), while researchers like Tiassou applied Discrete Event Simulation (DES) techniques to his Stochastic Activity Net (SAN) that described operational aspects of an aircraft. "A stochastic modeling formalism using the basic notions of place, marking and transition of Petri nets" (Tiassou, 2013, 56), helped Tiassou predict the operational reliability of an aircraft.

The formalism used by researchers such as Tiassou apply to extremely complicated Petri nets and Markov chains. They can be used to represent repairable and interdependent systems and can be simulated to determine system reliability, and in some cases their simulation may even be required. "For large series/ parallel structures, approximate expressions are known in the literature. For very large or complex system, a Monte Carlo simulation can become necessary" (IEC, 2006, 41). The technique known as Markov Chain Monte Carlo (MCMC) simulation is a technique used by some reliability researchers to randomly sample from a probability distribution based on a Markov Chain that is representative of the system under study.

The pitfalls of interpreting the results of a system's state-space representation is avoided by understanding the underlying assumptions of the models themselves and accounting for their limitations. As, according to IEC 61165, Petri nets are converted to their corresponding Markov model for analysis, Petri nets and Markov models are based on similar assumptions and suffer from the same limitations. The underlying assumption of a Markov chain is what is known as the Markov property. The principle of the Markov property is described by Boyd, a NASA reliability research scientist, as a situation where “the future behavior of the simplified stochastic process (i.e. Markov model) is dependent only on the present state and not on how or when the process arrived at that state” (Boyd, 1998, 7).

Assuming that the Markov property applies has many benefits, chief among them is “that it helps make the evaluation of Markovian models tractable. It is something of a mixed blessing, however, in that it is a very restrictive assumption that is not always consistent with the reality of real-world system behavior” (Boyd, 1997, 7-8). Boyd later identifies further simplifying assumptions that only apply to homogeneous Markov chain analyses; the state holding times are exponentially distributed and that the transition rates between states are assumed as constant (Boyd, 1998, 8). According to the IEC, “the assumption of constant failure rate is reasonably acceptable for components in many systems before the wear out period” (IEC, 2006, 23), but should be justified by the researcher.

If the assumption of a constant failure rate is justified for an item prior to wear out, the transition rates of repair – that is, the propensity of the unit to be repaired in the next small interval of time, given that the unit has failed prior to time t – should also be considered as carefully as failure rate. “If the assumptions are too inconsistent with the characteristics of the real system, then any dependability estimates…obtained from the model cannot be used to predict the behavior of the real system” (Boyd, 1998, 8). It is these assumptions, and the difficulty of creating Markov chains, that the IEC warns,

> The main problem is that the number of states and possible transitions increases rapidly with the number of elements in the system. The larger the number of states and transitions, the more likely it is that there will be errors and misrepresentations. (IEC, 2006, 23)

Despite these limitations, the power of discrete time Markov chains, whether homogeneous or non-homogeneous, to model the reliability and dependability of repairable systems should not be overlooked.

The preceding state-space reliability prediction techniques are applicable to the Low Cost Attritable Aircraft Technology demonstrator as it is assumed that the LCAAT system is a complicated system that: (1) consists of subsystems that are non-repairable and exhibit interdependent failure modes, (2) is an attritable, but not expendable, and is therefore intended to be pressed to operation after more than one use, and (3) is intended to be employed operationally for a short period of time, yet may still undergo all phases of lifecycle phases, including burn-in and wearout.

2.3.4 Reliability Model Development

While the preceding section examines the available reliability modeling techniques and their limitations, it is critical to any reliability study to take the necessary steps to properly develop the model. IEC 61165 outlines universal considerations for the development of a reliability model:

1) Set the goal of the analysis and define the unit of measurement

2) Define the system characteristics and the boundary of the analysis

3) Ensure that the choice in technique is the most appropriate for the task

4) Review the model and inputs with field practitioners (IEC, 2006, 29)

These considerations are similar to those tasks outlined by the Reliability Analysis Center's (RAC) process for Markov modeling as illustrated in Figure 7. RAC takes the guidance of IEC one step further by identifying when to use state reduction and simplification techniques as well as illustrating the iterative nature of reliability model development.

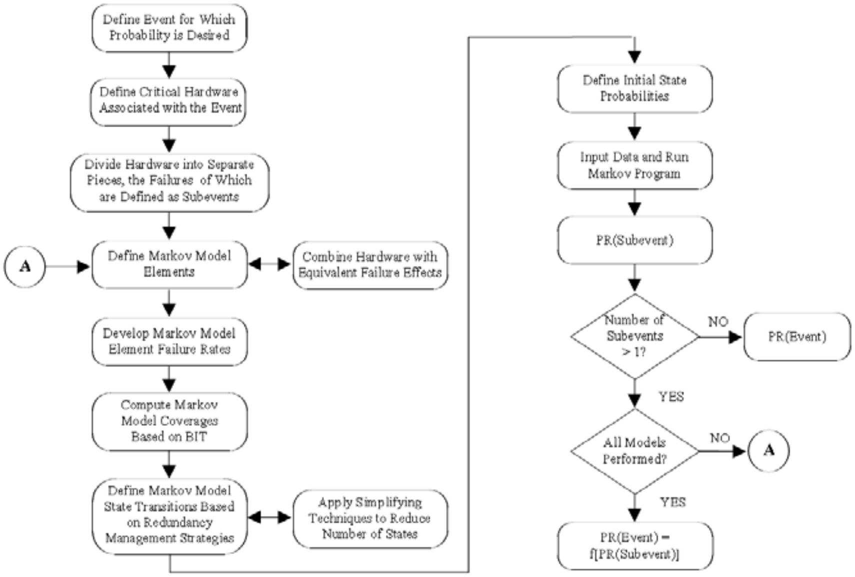

Define Event for Which Probability is Desired
Define Critical Hardware Associated with the Event
Divide Hardware into Separate Pieces, the Failures of Which are Defined as Subevents
A
Define Markov Model Elements
Combine Hardware with Equivalent Failure Effects
Develop Markov Model Element Failure Rates
Compute Markov Model Coverages Based on BIT
Define Markov Model State Transitions Based on Redundancy Management Strategies
Apply Simplifying Techniques to Reduce Number of States
Define Initial State Probabilities
Input Data and Run Markov Program
PR(Subevent)
Number of Subevents > 1?
NO
PR(Event)
YES
All Models Performed?
NO
A
YES
PR(Event) = f[PR(Subevent)]

Figure 7: Markov Modeling Process (Source: RAC, 2003)

The process outlined in Figure 7 may be specifically tailored to Markov techniques, but it is similar to the modeling processes used for other modeling techniques. Tiassou (2013) employed SAN formalisms when he created his model based on the Petri nets to predict the reliability of aircraft and his model construction process is illustrated in Figure 8. Figure 8 does not illustrate the iterative nature of reliability modeling like Figure 7, yet it is evident that model tuning with up-to-date data is critical to yielding accurate results.

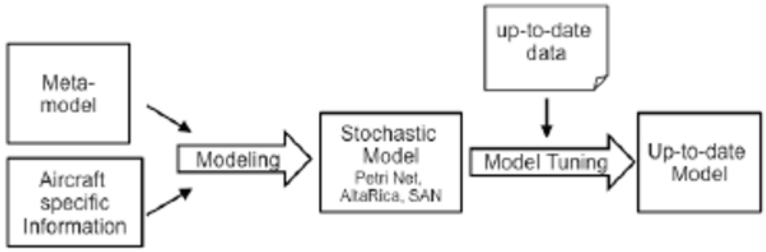


Figure 8: Operational dependability model construction process (Source: Tiassou, 2013)

The guidelines and processes outlined by the IEC, RAC, and researchers like Tiassou are intended to produce a reliability model that can yield meaningful predictions of the real behavior of a system. The next step in reliability analysis is to use the resultant model to meet the goals of the analysis. As the one of the stated goals of this research is to determine the effect that varying attritable system reliability and reparability has on the expected costs of a system, the researcher must also understand the implications of expected costs for and attritable system.

2.4 Cost Estimation and Cost Risk

The need for accurate cost estimation has led to the development of numerous techniques; they are tailored to a product's specific architecture and phase in the development cycle. DoD Instruction (DoDI) 5000.73 Cost Analysis Guidance and Procedures (9 June 2015) discusses the importance of employing the proper cost estimation technique, based on acquisition phase, data availability, and past experience. According to DoD guidance, cost estimation during early system development has an effect on the long-term success of the program. "At this formative stage, O&S cost considerations support the systems engineering process and influence requirements decisions followed by the system design decisions." (DoDI 5000.73). Just as Barnard indicated earlier, cost is intrinsically linked to systems engineering and is demonstrated by reliability engineering. Determining the proper cost estimation methodology for early reliability and cost trade studies remains the issue.

Unequivocal guidance on the proper methodology is found in DoD Directive 5105.84, Director of Cost Assessment and Program Evaluation, which identifies four general cost estimation methodologies: 1) Analogy, 2) Parametric (statistical), 3) Engineering, 4) and Actual Costs. The Defense Acquisition University (DAU) illustrates the recommended methodology, dependent on lifecycle phase, in Figure 9.

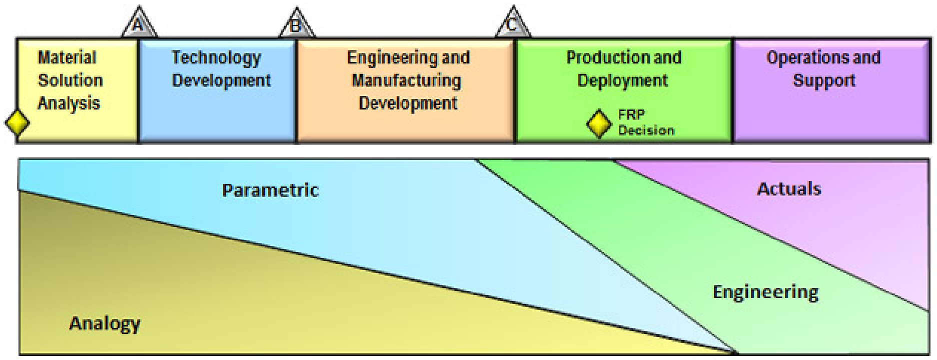


Figure 9: Cost Estimating Methodology (Source: DAU)

DAU advises that “the analogy method is most often used early in the program when little is known about the specific system to be developed. The parametric technique is useful throughout the program, provided there is a database of sufficient size, quality, and homogeneity to develop cost estimating relationships (CER)” (DAU, 2016). Given Figure 9, and the lack of cost data for an attritable air vehicle, the most applicable cost estimation technique the circumstances is the analogy method.

Industry has devoted much effort to characterize the effect of that changing reliability can have on cost, and in the instance of early systems they use they also use the analogy method to determine cost. For example, Jirutitijaroen and Singh (2004) develop a Markov model to trade design and maintenance parameters and study their effect on reliability (even though it is defined as MTTF) and cost. They then completed a sensitivity study on the effect of varying maintenance parameters such as time spent in each system state, inspection rate in each state, and probabilities of transition from each state. Their research also specifies an expression for system lifecycle cost as a function of failure cost, maintenance cost, and inspection cost. In their case, they argue that the expected total cost is a summation of expected failure and maintenance costs (Jirutitijaroen & Singh, 2004, p. 220). They defined expected failure cost, C_F, as:

$C_F = Failure\ Cost \times\ Failure\ Frequency$; similarly, expected maintenance cost, C_M, is defined as: $C_M = Maintenance\ Cost \times\ Maintenance\ Frequency$.

This concept of expected total cost as a summation of the impact of an event's occurrence and probability of its occurrence is referred to as another term, risk, in case the situation outlined by Jirutitijaroen and Singh, cost risk. "The notion of risk involves two concepts: (1) the *likelihood* that some problematic event will occur, (2) the *impact* of the event if it does occur. Risk is a joint function of the two; that is, $Risk = f\,(likelihood, impact)$" (Nicholas and Steyn, 2008, p.363). One half of the inputs to determine cost risk is found by using the aforementioned reliability models to determine the probability of failure and repair, also considered maintenance, through the multitude of reliability distribution functions. The other input, *impact*, can be determined by using the analogy cost estimation method suggested by the DoD and previously examined.

2.5 Competing Risks Analysis

The preceding simplified equation of risk is further complicated by what are known as competing risks, i.e. an event whose occurrence alters or even eliminates the probability of observing the event of interest (Freels, 2013). This is applicable to an attritable air vehicle as a failure caused by the launch subsystem may drive the probability of observing a failure of the electronics or recovery subsystems to zero. “In a competing risks framework once the system has failed due to risk $j = 1, \dots, J$ the probability of observing the system fail due to any of the remaining $J - 1$ risks is altered, and an informative censoring scheme is required” (Freels, 2013). However, without prior knowledge of each competing risk’s failure time distribution, non-parametric competing risk analysis must be used. According to Freels (2013), there are “two commonly used non-parametric estimation techniques in competing risks analysis: the complement of the Kaplan-Meier estimator $(1 - KM)$ and the Cumulative Incidence Function (CIF)” (Freels).

The complement of the Kaplan-Meier estimator method has been shown by Gooley et al. (1999) and Putter et al. (2007) to be imperfect as it overestimates the rate of occurrence of each event. Thus, Kalbfleish and Prentice's development of a CIF based on the Kaplan-Meier survivor function $KM_{12}(t) = KM_1(t) \times KM_2(t)$ where:

$$KM_1(t) = \prod_{t_i<t}(1-\frac{f_i}{n_i}) \text{ and } KM_2(t) = \prod_{t_i<t}(1-\frac{r_i}{n_i})$$

and Kalbfleish and Prentice's notation define the following:

$n \equiv the\ initial\ number\ of\ items\ at\ risk$

$f_i \equiv the\ number\ of\ items\ failed\ from\ the\ event\ of\ interest\ prior\ to\ time\ t_i$

$r_i \equiv the\ number\ of\ items\ failed\ from\ the\ competing\ risk\ prior\ to\ time\ t_i$

$n_i \equiv the\ number\ of\ items\ at\ risk\ after\ time\ t_i$

is expressed as:

$$CIF(t) = \sum_{i=1}^{s} \frac{f_i}{n_i - 1} \times KM_{12}(t_i) \qquad [2]$$

Where s is the largest i, such that $t_i < t$.

The CIF is useful as an input to reliability models like Markov chains as it "is a function of the hazard rates for both modes making CIF the preferred method of estimating the failure probability when competing risks are present" (Freels, 2013).

2.6 Gap Analysis

This review of the literature demonstrates the importance of defining reliability in terms of operational success according to a given concept of operation. Explanations for the successful employment, as well as the limitations, of modeling reliability techniques and cost estimation methods are were also discussed. This literature has made great contributions to the field of reliability engineering, as it forms the basis for reliability engineering of an attritable air vehicle like the Low Cost Attritable Aircraft Technology demonstrator. Employing the previously discussed reliability modeling techniques, allows for the determination of an attritable system's sensitivity to variations in reliability and reparability, as well as its effect on the cost risk of changing these parameters. If used correctly, this information serves as a tool to decrease the acquisition and lifecycle costs of attritable air vehicles desired by AFRL researchers.

3.0 Methodology to Trade Attritable System Reliability and Cost

3.1 Chapter Overview

The objective of the Low Cost Attritable Aircraft Technology (LCAAT) demonstration is to trade “the relatively high cost of UAV performance, design life, reliability, and maintainability” (military aerospace) to achieve a re-usable system “whereby virtue of its cost, the loss of the aircraft could be tolerated” (AFRL/RQKP, 2015, p.1). According to AFRL, system attributes like design life, reliability, and maintainability drive lifecycle costs and therefore, “need to be traded to achieve the optimum capability/cost effects” (AFRL/RQKP, 2015, 1). This research investigates reliability modeling techniques that are applicable to attritable air vehicles. The most appropriate technique is used to determine the sensitivity of a similar system, known as the baseline system, to variations in subsystem reliability and reparability by examining that system’s failure-time data. Finally, the impact that these trade have on cost risk is also calculated as a system’s attritability is a “virtue of its cost” (AFRL/RQKP, 2015, p.1).

This chapter outlines three important phases necessary to ensure this research of trading reliability and reparability for lower cost is bounded, tractable, and repeatable: an outline of the simplifying assumptions made to ensure consistency in the analysis of the baseline system’s failure-time data and streamline the representation of the system under study; the failure-time data is illustrated and the analysis tools are identified; and a step-by-step description of the analysis activities is provided. These activities use the failure-time data gathered on a baseline system and analysis tools as inputs for the investigation of reliability and reparability trades.

3.2 Assumptions

According to a 1957 report by the Advisory Group on the Reliability of Electronics Engineers (AGREE), reliability is defined as the "probability that a system will perform in a satisfactory manner for a given period of time when used under specified operating conditions in a given environment" (Hogge, 2012, p.8). Accordingly, the calculation of system reliability only applies to a set instance of satisfactory performance, operating condition, and given environment. To gain valuable results from reliability estimation, the parameters of reliability, reparability, and cost risk must be accurately defined.

In the case of unmanned air vehicles, satisfactory system performance is commonly considered the successful performance of its mission, without subsystem failure, in a manner that allows for its continued operation for the length of its intended lifecycle. However, the application of this assertion to an attritable air vehicle is met with many challenges. The intended length of the attritable air vehicle's lifecycle is as yet undetermined and therefore, no assumptions can be made regarding the period of time that the system must perform satisfactorily. Likewise, the expected mission length is undetermined as the system must perform in a range of threat environments – not only experiencing removal from the population due to system-level events but also due to external factors.

Additionally, the operational concept of the vision system may prohibit the repair of certain subsystems. For the purposes of failure-time data analysis, it is assumed that the failed systems that undergo repair are not repaired to an "as good as new" standard, and are instead restored to the standard of "as bad as old" – meaning that only the failed subsystem is repaired or replaced upon failure and the entire system does not undergo full restoration.

The aforementioned baseline vehicle, upon which failure-time data is collected, already operates in a range of threat environments for the DoD. As the vision LCAAT system leverages non-developmental subsystems readily available for production, the study of this baseline vehicle provides useful insight into the future reliability and cost risk of LCAAT. This research assumes that the LCAAT system and the existing system are sufficiently similar that investigation of reliability for the underlying physical architecture is appropriate to use as a baseline. The use of this baseline make the calculation of the impact of trading reliability and reparability tractable. Figure 10 displays a block definition diagram (bdd) that illustrates the physical architecture of the baseline air vehicle. It is important to note that this research does not consider the reliability of the mission payload as it does not affect the performance of the vehicle. Omitting the mission payload as a part of the attritable air vehicle architecture simplifies the application of the baseline reliability data to the LCAAT vision system.

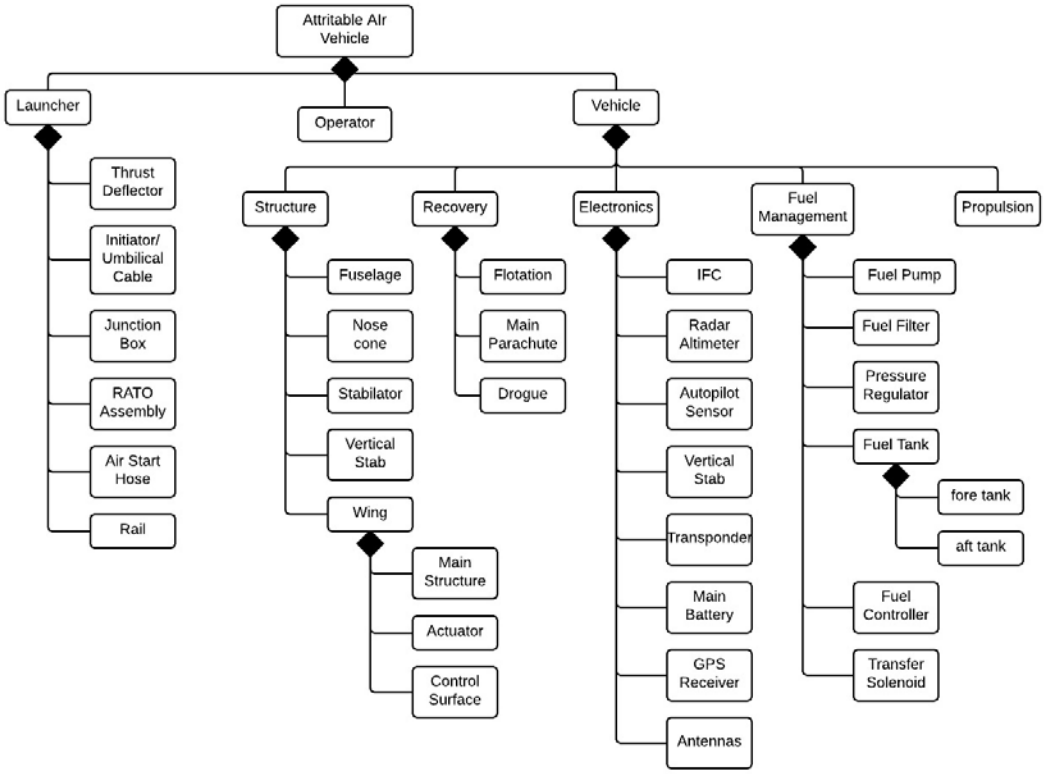


Figure 10: Attritable Air Vehicle UML Object Diagram (Note: System representation excludes mission payload)

The seven primary subsystems of the baseline system are illustrated in Figure 10. They include: the electronics, fuel management subsystem, launcher subsystem, operator, propulsion subsystem, recovery subsystem, and structural subsystem. The baseline system's failure-time data only outlines the failure behavior at the subsystem level of abstraction consistently. Reliability analysis at the component level, a more detailed level of abstraction, is prohibitively complicated without this data. Therefore, this research is limited to trading the reliability and reparability of these seven previously mentioned subsystems. These subsystems are consistent with the published DoD guidance outlined by MIL-STD 881C, Appendix H dated 14 Jan 2011. This guidance provides a generic framework of a Work Breakdown Structure (WBS) of Unmanned

Aerial Vehicles (UAVs). The concordance of the physical architecture is advantageous as DoDI 5000.02 states that a WBS is used to track costs for all defense acquisitions programs.

It necessary to discuss the assumptions of the reliability estimation model as this research's objective is to model the impact reliability and reparability variations on system survivability and cost risk. This research employs the Markov chain technique to represent the behavior of a system as a state transition diagram. Specifically, this research builds discrete time-nonhomogeneous Markov chains to allow for the variation hazard and repair rates over time, according to a global clock. A single Markov chain represents a system as elements "which can assume only one of two states: up or down. The system as a whole, however, can assume many different states, each being determined by the particular combination of functioning and failed elements" (IEC 61165, 2006, p.21).

The IEC gives guidelines for the application of Markov techniques to practical situations and stipulates that the number of possible states must be finite and that the sum of all states probabilities must also be unity, "i.e. at any instant in time, the system can be in one and only one of the states in the state transition diagram" (IEC 61165, 2006, p.23). This standard also outlines how a system that includes non-restorable elements, like those in the baseline air vehicle, can be specified. Such a system "can be regarded as a special case of a system with restorable elements where the restoration rates are zero (or restoration times are infinite)" (IEC 61165, 2006, p.23).

The Markov technique is hampered by what Boyd, a NASA reliability engineer terms a restrictive assumption: the Markov property. According to this property, "the future behavior of the simplified stochastic process (i.e. Markov model) is dependent only on the present state and not on how or when the process arrived at that state" (Boyd, 1998, 7). Thus, a homogenous Markov chain concludes that "the state holding times are exponentially distributed and the transition rates between states are assumed as constant" (Boyd, 1998, 8). However, "a non-homogeneous Markov chain is characterized by the fact that the transition rates between states are functions of a global clock, e.g. elapsed time" (RAC, 2003, p. 2). Therefore, the employment of discrete time-nonhomogeneous Markov chains by this research allows for the definition of time dependent transition rates, where the transition rates represent either the hazard rate or the repair rate. This allows for the examination of system reliability and reparability throughout the lifecycle, instead of only during its useful life when its hazard rate is assumed to be constant.

Studies of the reliability and reparability for the baseline vehicle are complicated by the complicated censoring scheme and competing failure modes that fundamentally alter the probability of observing a particular failure mode. The implications of censoring and competing risks on failure-time data, as well as mitigating techniques, are outlined in the subsequent section on the specifics of the gathered data.

3.3 Data and Materials

The performance, reliability, and system cost data on a baseline air vehicle in the DoD are analyzed to determine the effect of varying subsystem reliability and reparability on system-level reliability and cost risk. The System Performance Document outlines the performance of the system and also as serves as the requirements document. A catalog of mission performance, dating back to program inception, provides information on unit launches, failure mode, and the root cause for over 1100 sorties. This catalog is shown to be complete enough to cross-reference a separate summary of system failures, and also shows that the population of over 300 units are comparable enough to combine into a single homogeneous population.

This data is used to determine each subsystem's time to failure, based on the number of cycles the system underwent, as well as the repair strategy used at the time. However, there are complications present in these reliability data. The random removal of units from the population present a unique censoring mechanism that obscures the observation of each subsystem's exact failure time. Occasionally, external events that are a function of the operating environment result in the removal of the unit from the population. This prohibits the observance of the exact failure time of the unit. This type of censoring is referred to as Type I, time, or right censoring and must be accounted for in the calculation of reliability distribution functions.

The failure-time data on the baseline attritable air vehicle offers an added complexity, the presence of competing risks. "A competing risk is an even whose occurrence fundamentally alters or altogether eliminates the probability of observing the event of interest" (Freels, 2013). The simplest manifestation of a competing risk for an attritable air vehicle is a launch subsystem failure that drives the probability of observing a failure of any other subsystem to zero. According to Freels, "the cumulative incidence function (CIF) estimator is a function of the hazard rates for both modes making CIF the preferred method of estimating the failure probability when competing risks are present" (Freels, 2013).

To efficiently calculate the CIF of each respective risk, heretofore defined as the failure of a given subsystem, this research uses the 'cmprsk' package (Gray, 2015) developed for use in R 2.2-7 or later. R is an open source language and environment that is tailored statistical and graphical analysis of data. This package employs the previously defined equations to calculate CIF. Additionally, as "the cumulative incidence function (CIF) estimator is a function of the hazard rates for both modes" (Freels, 2013), it is also possible to determine the hazard rate of a given failure mode using the subdistribution hazard technique (Gray, 1988).

The hazard rate, $h(t)$, is defined by Meeker and Escobar (1998) as: $h(t) = \frac{f(t)}{1-F(t)}$.

Yet, the output of the 'cmprsk' function shows that the CIF is directly related to the complement of reliability function and represents the probability of a unit's failure before time t. Thus, it can be said that the hazard function $h(t)$ is represented as $h(t) = \frac{CIF'(t)}{1-CIF(t)}$ when competing risks are present. Furthermore, as the failure-time data of the baseline air vehicle only yielded failure information at discrete time intervals, i.e. at the occurrence of each sortie, n, and therefore the derivative of the cumulative incidence function is denoted as $CIF'(n) = CIF(n+1)$ - $CIF(n)$. Therefore, this research defines the hazard function as $h(n) = \frac{CIF(n+1) - CIF(n)}{1-CIF(n)}$. The hazard function is an input to the discrete time-nonhomogenous Markov chain models built to determine the probability of failure over time.

In addition to using the 'cmprsk' package developed to calculate the CIF estimator for situations where competing risks are present, this research also leverages the ‘markovchain’ package (Spedicato et al., 2016). Developed to perform statistical analysis for discrete time Markov chains, this research uses the markovchainList class to create a state-transition matrix for each discrete sortie up to the point where there is not enough data to create an accurate Markov chain. According to Meeker and Escobar, the data necessary to construct a state-transition matrix is dependent upon the availability of a “large” number of samples. “With censored data…a typical guideline for large is 20 or more, but this really depends on the problem and the questions to be answered” (Meeker & Escobar, 1998, p. 36). Therefore, further analysis of the baseline failure-time data will only extend to the ninth sortie due to the lack of systems in the population at any additional interval of time.

The baseline air vehicle failure-time data is presented in Table 1. This table presents the number of units in the population at each given sortie – note that the most sorties undergone by a unit is 19. Furthermore, the table presents the failure mode, i.e. the subsystem determined to be the root cause of a failure to perform satisfactorily, as well as the number of units destroyed after the occurrence of a given failure mode.

Table 1: Baseline System Failure-time Data

		Sortie Number																		
Failure Mode	**Sortie Outcome**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
Electronics Subsystem	Total	9	14	4	5	5	3	2	2	1	1	0	0	0	0	0	0	0	0	0
	Destroyed	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fuel Management Subsystem	Total	13	7	5	6	3	3	1	0	1	2	0	0	0	0	0	0	0	0	0
	Destroyed	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
Launcher Subsystem	Total	8	5	3	5	1	1	1	1	1	0	0	0	0	0	0	1	0	0	0
	Destroyed	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Operator Subsystem	Total	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
	Destroyed	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Propulsion Subsystem	Total	3	6	4	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
	Destroyed	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Recovery Subsystem	Total	4	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
	Destroyed	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Structure Subsystem	Total	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Destroyed	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Type I Censored		275	194	148	97	70	44	32	24	19	12	12	7	6	5	4	2	2	1	1
Total in Population		313	227	227	116	79	52	37	29	22	15	12	7	6	5	4	3	2	1	1

Analysis of Table 1 uses the ‘cmprsk’ (Gray, 2015) and ‘markovchain’ (Spedicato et a., 2016) packages, available open source for R, to determine the sensitivity of an attritable system’s reliability to changes in subsystem reliability and reparability. The ‘cmprsk’ package produces reliability density functions that are inputs to discrete time-nonhomogeneous Markov chain lists built using the ‘markovchain’ package. The state probabilities produced by these analyses are then input to calculations of cost risk per sortie. The detailed methodology for determining these attributes is outlined in the subsequent section.

3.4 Processes and Procedures

The order in which the aforementioned analyses are accomplished is important to the overall results of a reliability trade study. The activity diagram depicted in Figure 11 illustrates the sequence of activities necessary to determine the effect of trading reliability and reparability for an attritable system. The activity diagram is separated into "swim lanes" to signify the division of responsibilities based on the subject matter of each respective activity. Reliability-related analyses include: the analysis of baseline air vehicle failure-time data, the variation of attritable air vehicle subsystem reliability – represented by its hazard rate, $h(n)$ – and reparability, as well as the construction and analysis of the resultant discrete time-nonhomogeneous Markov chain models. The resulting reliability probability functions are used in conjunction with the cost-related activities of repair and maintenance cost parameterization to finally calculate cost risk.

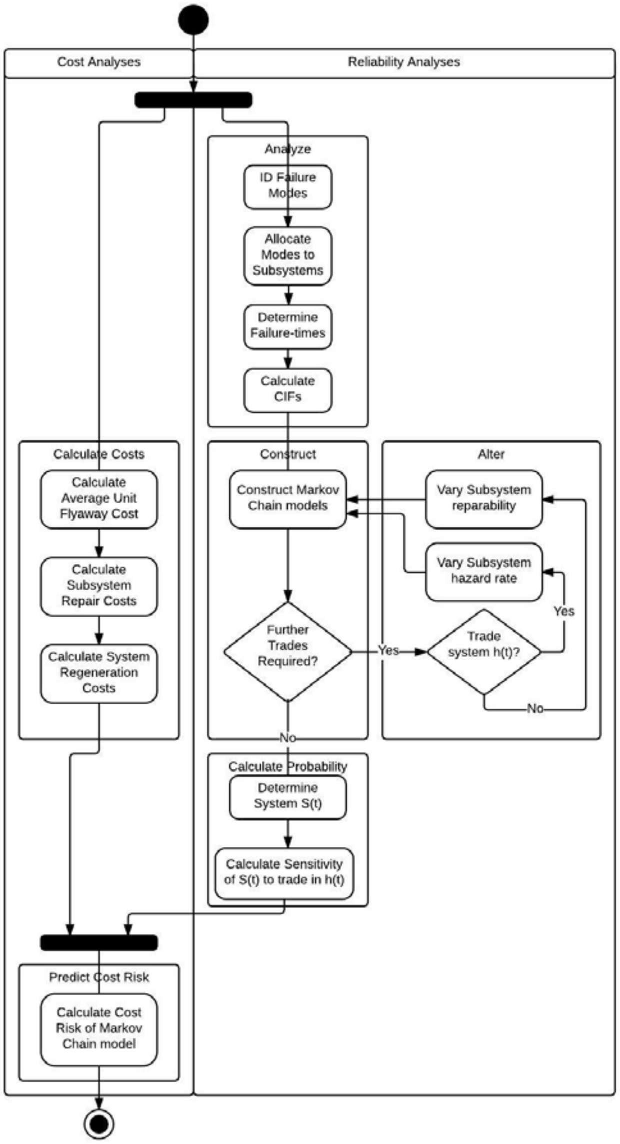

Cost Analyses
Reliability Analyses
Analyze
ID Failure Modes
Allocate Modes to Subsystems
Determine Failure-times
Calculate CIFs
Calculate Costs
Calculate Average Unit Flyaway Cost
Calculate Subsystem Repair Costs
Calculate System Regeneration Costs
Construct
Construct Markov Chain models
Further Trades Required?
Yes
No
Alter
Vary Subsystem reparability
Vary Subsystem hazard rate
Trade system h(t)?
Yes
No
Calculate Probability
Determine System S(t)
Calculate Sensitivity of S(t) to trade in h(t)
Predict Cost Risk
Calculate Cost Risk of Markov Chain model

Figure 11: Methodology for the Study of Reliability and Cost Trades for Attritable Systems

The first phase of Figure 11 describes the approach to analyze data of a baseline fielded attritable air vehicle. In this analysis, the failure modes that impede satisfactory system performance are identified. Determination of root cause allows for the allocation of these failure modes to the subsystem at fault. Lastly, the calculation of the hazard rate of each subsystem's cumulative incidence function is used to more accurately estimate the hazard rate given the existence of competing risks, or failure modes.

The construction phase includes the creation of a reliability model in the form of a discrete time-nonhomogeneous Markov Chain model. A discrete time-nonhomogeneous model allows for the variation of the transition probabilities according to a global clock. Therefore, a time-nonhomogeneous model is considered a collection of homogeneous Markov chains that describe system behavior at discrete points in time. Figure 12 illustrates the state-transition diagram of the Markov chains used to create the discrete time-homogeneous Markov Chain models used in this research and presented in Appendix A. Note the existence of nine states: operational, destroyed (the absorbing state), and the failed states based on the seven subsystems previously identified.

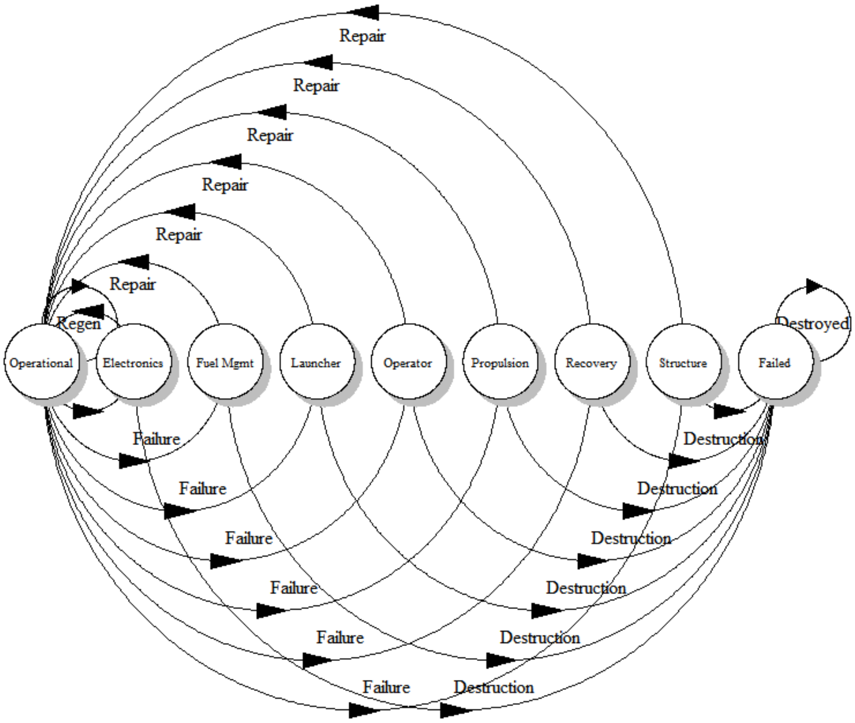


Figure 12: Example Markov Chain Model with Identified Subsystems

Altering the baseline discrete time-nonhomogeneous Markov Chain model is accomplished by varying each respective subsystem's competing risk hazard rate, as well as varying each subsystem's reparability – that is, the rate at which the system is restored to an operational state from a failed state. These variations allow for the determination of the sensitivity of system-level reliability to these changes. The decision of which reliability and reparability trade excursions to make considers the realistic direction of change based on the existence and behavior of similar subsystems in the field.

Table 2: Direction of Subsystem Hazard Rate Variation and Justification

Subsystem	**Direction of Variation**	**Justification**
Electronics	▲5%	the effect of trading of comprehensively tested MIL-SPEC electronics for COTS computers and sensors (with possibly lower reliability) is unknown
Fuel Management	▲5%	the effect of trading of comprehensively tested MIL-SPEC components for COTS tanks, filters, and pumps (with possibly lower reliability) is unknown
Launcher	▼5%	The effect of trading of the comparatively high risk method of zero-length launch methods for more conventional methods (with lower risk) is unknown. *(This trade excursion assumes that the rest of the vehicle's subsystems remain unchanged – a simplifying assumption as it would require reconfiguration of the airframe if implemented)*
Operator	▼5%	It is unlikely that operator performance could be traded to lower cost as policy dictates that the need for skilled operators overrides the training costs. Therefore, the effect of increasing operator reliability, through means such as training, is explored
Propulsion	▼5%	The impact of trading the baseline attritable system's jet engine which are also used on expendable weapon systems for FAA-certified engines (with possibly more reliability) is unknown
Recovery	▼5%	The effect of trading of the comparatively high risk method of parachute recovery for more conventional landing methods (with lower risk) is unknown. *(This trade excursion assumes that the rest of the vehicle's subsystems remain unchanged – a simplifying assumption as it would require reconfiguration of the airframe if implemented)*
Structure	▲5%	Research is ongoing into the effect of trading of comprehensively tested MIL-SPEC structural components for rapidly manufactured component for COTS Computers and Sensors could decrease cost, but also reliability

Table 3: Alteration of Subsystem Reparability and Justification

Subsystem	Level of Baseline Reparability	Level of Altered Reparability	Justification
Electronics	reparable	no change	Air vehicle onboard electronics constitute a considerable portion of unit cost, yet this subsystem's repair-to-buy cost ratio is considerably low. The impact of this trade is not the subject of ongoing research.
Fuel Management	reparable	no change	The repair-to-buy cost ratio is relatively high for the Fuel Management Subsystem as repairs generally consist of replacement. However, this subsystem constitutes a very small portion of unit cost. The impact of this trade is not the subject of ongoing research.
Launcher	reparable	no change	Whether the system is reparable after a launch subsystem failure is dependent upon the impact to the vehicle as the launcher is separate from the vehicle (see Figure 10). The impact of this trade is not the subject of ongoing research.
Operator	reparable	no change	Whether the system is reparable after an operator error is dependent upon the impact to the vehicle as the operator is separate from the vehicle (see Figure 10). The impact of this trade is not the subject of ongoing research.
Propulsion	reparable	non-repairable	The propulsion subsystem represents a high percentage of unit cost as well as a high cost to repair (instead of replace). The impact of this trade is the subject of ongoing research.
Recovery	reparable	no change	The repair-to-buy cost ratio is relatively high for the Recovery Subsystem as repairs generally consist of replacement. However, this subsystem constitutes a very small portion of unit cost. The impact of this trade is not the subject of ongoing research.
Structure	reparable	non-repairable	The system structure, or airframe, represents a high percentage of unit cost as well as a high cost to repair (instead of replace). The impact of this trade is the subject of ongoing research.

Tables 2 and 3 show the feasible trades of reliability and reparability possible when considering the existence and realistic behavior of similar subsystems in the field. These tables also provide insight into the motivations for each trade excursion. The excursions outlined by these tables are considered in the construction of the appropriate discrete time-nonhomogeneous Markov Chain model as well as the determination of the system level reliability function for each model.

Last, but certainly not least, the final activity in Figure 11 is the prediction of absolute cost risk for each of the discrete time-nonhomogeneous Markov chain models. This activity uses example regeneration, repair, and average unit flyaway cost (AUFC) to veil proprietary cost information but also allow for the calculation of absolute cost risk with example changes in subsystem costs. Finally, these parameters (as well as the system level reliability function ascertained from the aforementioned Markov Chain models) are used to calculate system cost risk. The system cost risk is based off of Nicholas and Steyn's (2008) description of risk as the sum of the impact of an event multiplied by its probability of occurrence. The example cost parameters are presented in Table 5 and are based on Last Repair Cost (LRC) to Last Acquisition Cost (LAC) ratio data for a buy of 100 Unmanned Aerial Vehicles (UAVs) in FY2017.

Table 4: Example Cost Values Used in Cost Risk Estimation

Cost Category	Example Cost ($)	Repair-to-Buy Cost Ratio (%)
$Regeneration\ Cost$	45,000	
$Repair\ Cost_{electronics}$	1,000,000	15
$Repair\ Cost_{fuel\ mgmt}$	50,000	32
$Repair\ Cost_{launcher}$	540,000	12
$Repair\ Cost_{operator}$	N/A	N/A
$Repair\ Cost_{propulsion}$	600,000	22
$Repair\ Cost_{recovery}$	30,000	20
$Repair\ Cost_{structure}$	750,000	32

Thus, the absolute system cost risk at a given interval of time is defined as:

$$= \mathit{Regeneration\ Cost} \times P_{regeneraion}(t) + \sum_{i=\mathit{Subsystem\ 1\ Failure}}^{\mathit{Subsystem\ 7\ Failure}} (\mathit{Regeneration\ Cost} + \mathit{Repair\ Cost}_i \times P_{repair}(t)) + \mathit{AUFC} \times F(t)$$

where,

$\mathit{AUFC} \equiv \mathit{Average\ Unit\ Flyaway\ Cost}$

$\mathit{Regenerate\ Cost} \equiv \mathit{cost\ to\ prepare\ an\ operational\ system\ for\ another\ sortie}$

$P_{regeneration}(t) \equiv \mathit{probability\ of\ transition\ to\ operational\ state\ without\ failure}$

$\mathit{Repair\ Cost}_i \equiv \mathit{subsystem\ cost\ of\ repair,} (\mathit{Repair} - \mathit{to} - \mathit{Buy\ ratio} \times \mathit{Subsystem\ Cost})$

$P_{repair}(t) \equiv \mathit{probability\ of\ transition\ to\ each\ respective\ failure\ state}$

$F(t) \equiv \mathit{probability\ of\ transition\ the\ absorbing\ state\ of\ system\ failure,} 1 - S(t)$

For the purposes of this research, the cost to regenerate the baseline attritable system is considered as the sum of the cost of the consumables used throughout each sortie and the cost of labor required to recover, maintain, and pre-launch prepare for the next sortie.

The aforementioned failure-time data on the baseline system are presented in preceding discourse. The analyses of these data and how they address this research's investigative questions are presented in proceeding sections and follow the outlined methods to develop, evaluate, and modify reliability models of the baseline system. The impacts of changing reliability and reparability of the baseline system on cost risk are also presented in proceeding sections.

4.0 Data Analysis and Results

4.1 Chapter Overview

The interpretation of the time to failure data of the baseline air vehicle provides the basis by which this study trades reliability, reparability, and cost. This baseline system is fielded by the Department of Defense (DoD) and has accomplished over 1100 sorties over more than a decade to date. This failure-time data exhibited two characteristics that complicated the determination of system reliability, and the calculation of the system's sensitivity to changes in reliability: a complicated censoring scheme and competing failure modes. Both characteristics inherently alter the probability of observing a particular failure mode. This research employs the hazard rate of the cumulative incidence function (CIF) as an input into discrete time non-homogeneous Markov chain models to account for these complicating factors, and decrease the limitations of simplifying assumptions.

This chapter addresses the three investigative questions previously outlined regarding: (1) the suitability of reliability estimation methods for attritable system studies, (2) the sensitivity of attritable system reliability to changes in subsystem reliability and reparability, as well as (3) the effect of changing subsystem reliability and reparability on cost risk. First, the necessity of the discrete time-nonhomogeneous Markov chain model to fit the data is addressed. Next, the impact that changing hazard rate of the CIF, $h(t)$, and reparability (in accordance with Table 2 and 3) has on the probability of survival is examined. Finally, the cost at risk is determined by using the previously determined probabilities in addition to regeneration, repair, and system replacements costs.

4.2 Suitability of the Markov Chain Technique

The failure-time data of the baseline system is analyzed to determine the suitability of the discrete time-nonhomogeneous Markov chain modelling technique to the study of attritable air vehicles. For this technique to be suitable for the analysis of this data, the data must be shown to be discrete (i.e. not continuous across time) and time-nonhomogeneous (i.e. changing its behavior over time). Several characteristics point to the discrete nature of the baseline attritable air vehicle's failure-time data. As performance data were only consistently collected at discrete intervals (i.e. at the beginning and end of each sortie) and not continuously during operation, the failure-time data are proven to be discrete. Therefore, further computation of system reliability over time is calculated in terms of n, number of sorties undergone by the system.

Further examination of the discrete and non-constant nature of the failure behaviors is accomplished by studying the CIF of each subsystem. The failure-time data exhibits both censoring and competing risks factors, and Figure 13 represents each subsystem's CIF as a step function. The calculation of subsystem CIF is the first step to determine the hazard rate of each subsystem – that is, the propensity of the unit to fail in the next small interval of time, given survival to that time t (Meeker & Escobar, 1998, 28) – and thereby determine if whether the system's behavior changes with respect to time.

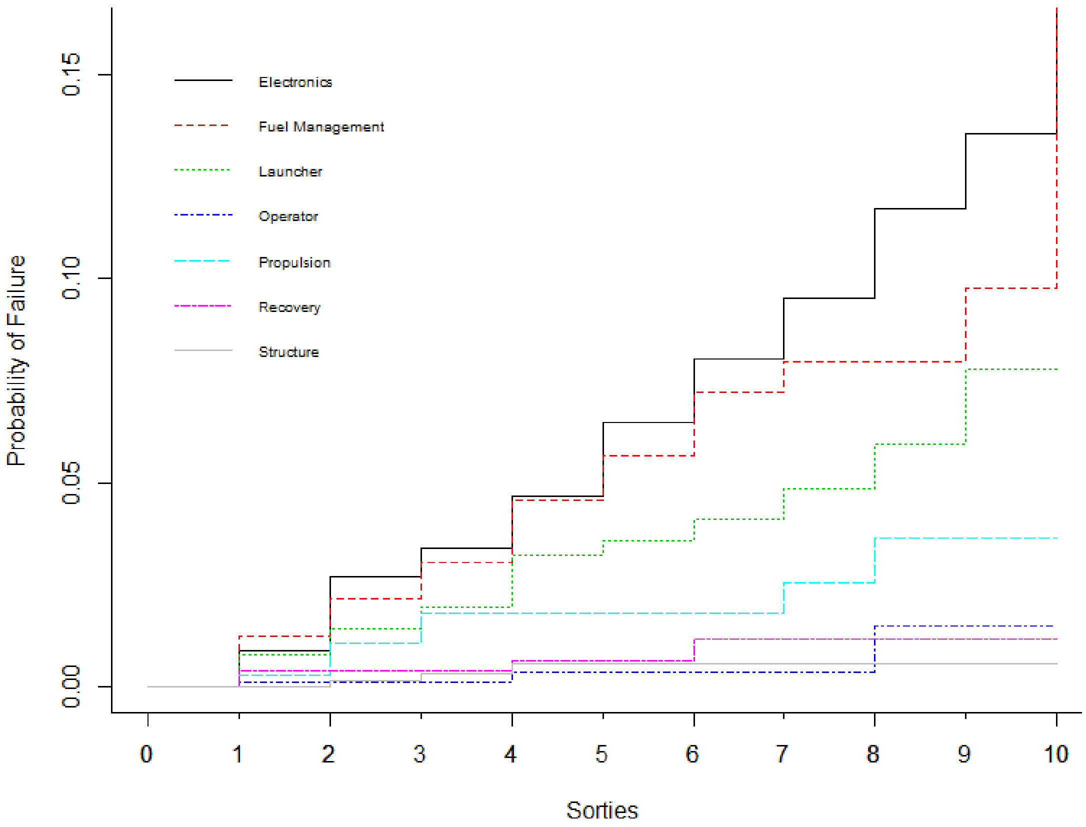


Figure 13: Cumulative Incidence Function (CIF) of Subsystem Failure Modes

To illustrate the system susceptibility to failure from a specific failure mode over the next time interval, previously defined as n, the hazard rate of the discrete CIF is: [4]

$$h(n) = \frac{CIF(n+1) - CIF(n)}{1 - CIF(n)}$$

and can be calculated for each respective subsystem. Figures 14-20 illustrate the changing nature of subsystem hazard rates over time with a 95% confidence interval. This confidence interval uses the variance estimate computed in Gray's 'cmprsk' (2014) that is based on the estimate of the asymptotic variance of Aalen (1978). The interval is based off of the logit transformation presented in Meeker and Escobar (1998, p.56) to increase the confidence interval to make it strictly positive. This transformation is implemented using the following equation: [5]

$$[C\underset{\sim}{I}F(n_i), \widetilde{CIF}(n_i)] = \left[\frac{\widehat{CIF}}{\widehat{CIF} + (1-\widehat{CIF}) \times w}, \frac{\widehat{CIF}}{\widehat{CIF} + (1-\widehat{CIF})/w}\right]$$

where $w = \exp\{z_{(1-\alpha/2)}\widehat{se}_{\widehat{CIF}}/[\widehat{CIF}(1-\widehat{CIF})]\}$

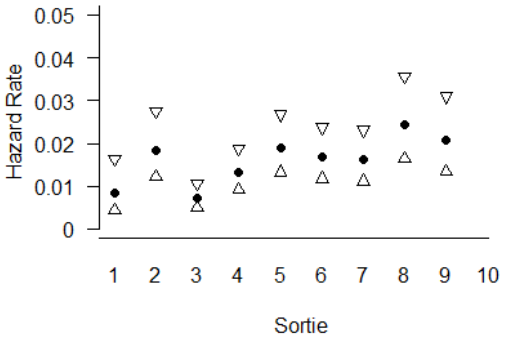


Figure 15: Hazard Rate of Electronics Subsystem

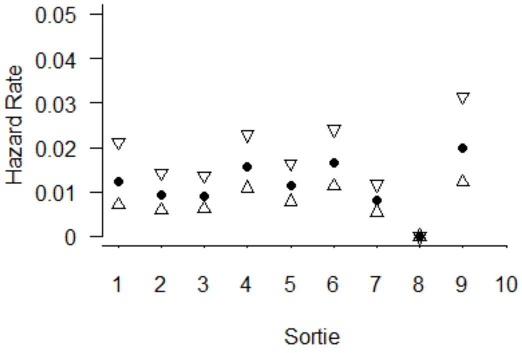


Figure 14: Hazard Rate of Fuel Mgmt Subsystem

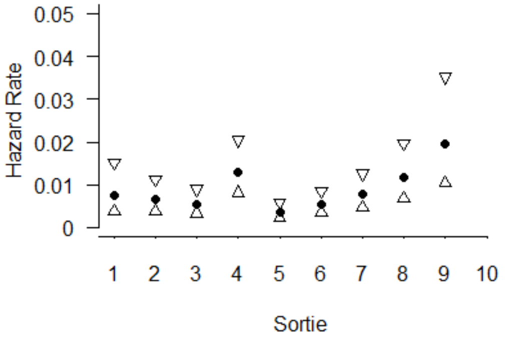


Figure 17: Hazard Rate of Launcher Subsystem

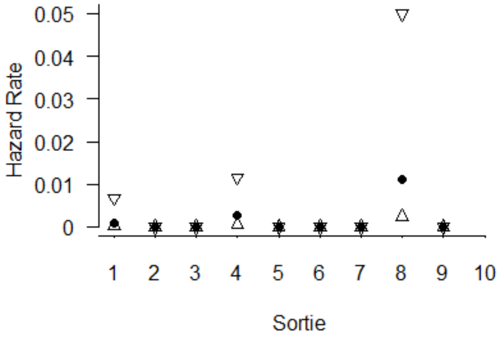


Figure 16: Hazard Rate of Operator

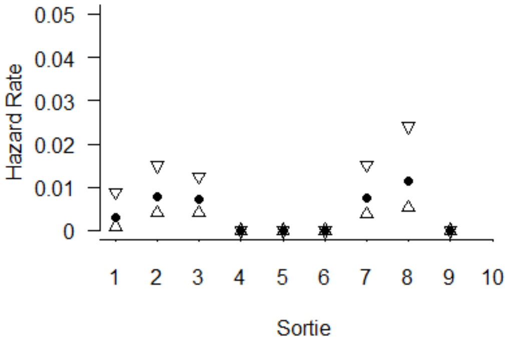


Figure 19: Hazard Rate of Propulsion Subsystem

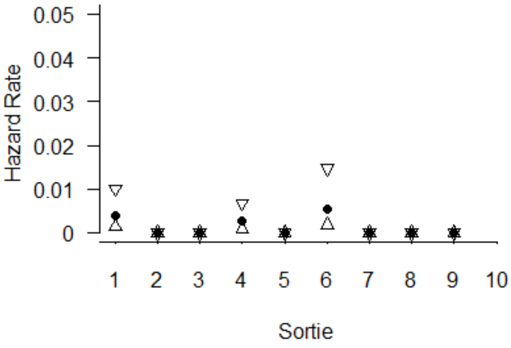


Figure 18: Hazard Rate of Recovery Subsystem

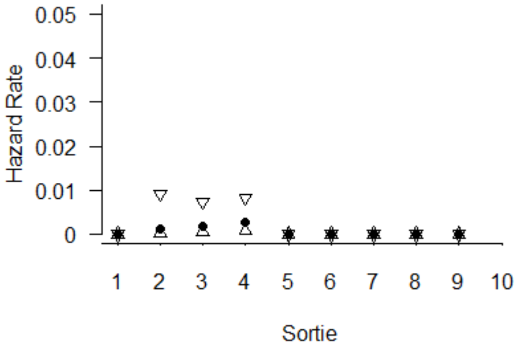


Figure 20: Hazard Rate of Structural Subsystem

The non-continuous and non-constant nature of each respective subsystem's hazards rates are shown Figures 14 - 20. The discrete nature of subsystem hazard rate is demonstrated in Figures 15,17,18,19, and 20 where subsystems experience no hazard during some sorties. The time-nonhomogeneous nature of subsystem hazard rates is shown as it is infeasible to assign a constant hazard rate that lies between the 95% confidence bounds at every interval in time and for each subsystem. Therefore, it can be said that the subsystem failure mode hazard rates are discrete and – at least graphically – change over time. Yet still, further statistical analysis is necessary to determine if the failure-time data could be reasonably simplified to not change with respect to time.

To test this hypothesis graphically a q-q, also referred to as a quantile-quantile, plot is used to compare two probability distributions by plotting each distribution's quantiles against each other. A simplification of the failure time data would assume that the hazard rates remain constant across all time. To test the validity of this assumption the sampled quantiles of the CIF data are plotted against the theoretical quantiles of the exponential distribution, as this distribution follows this simplifying assumption. The q-q plots of these data sets are illustrated in Figures 21-27. Graphically, subsystems with failure-time data points that do indeed follow the exponential distribution fall approximately on the quantile-quantile line in blue that represents the points at which the sample quantiles and theoretical quantiles (in this case, the quantiles of the exponential distribution) are equal.

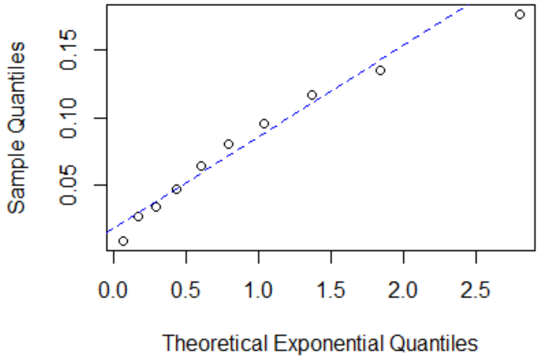


Figure 22: Electronics CIF vs. Exponential Q-Q Plot

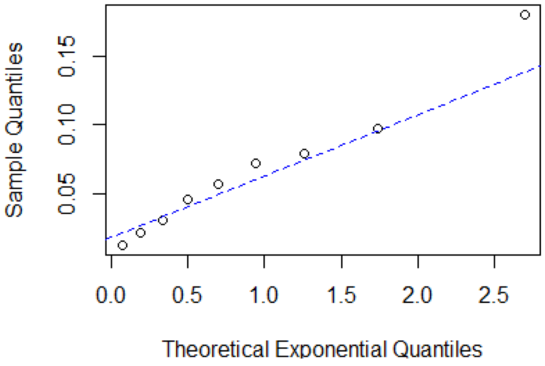


Figure 21: Fuel Mgmt CIF vs. Exponential Q-Q Plot

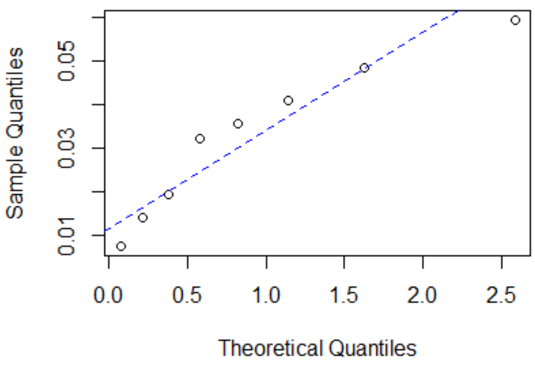


Figure 24: Launcher CIF vs. Exponential Q-Q Plot

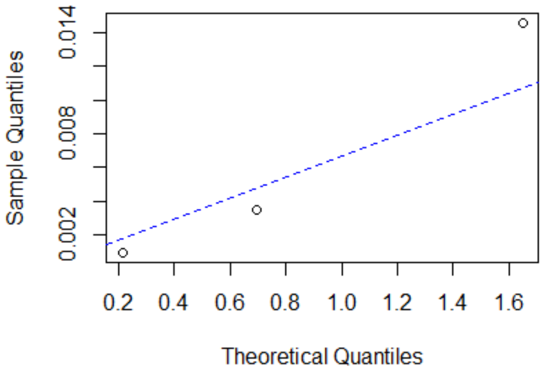


Figure 23: Operator CIF vs. Exponential Q-Q Plot

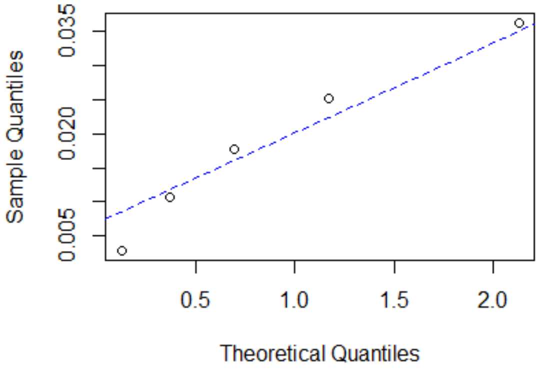


Figure 26: Propulsion CIF vs. Exponential Q-Q Plot

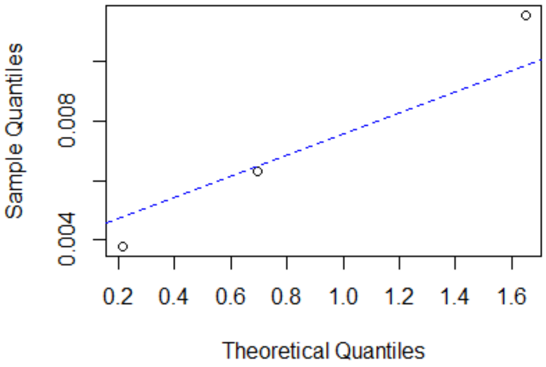


Figure 25: Recovery CIF vs. Exponential Q-Q Plot

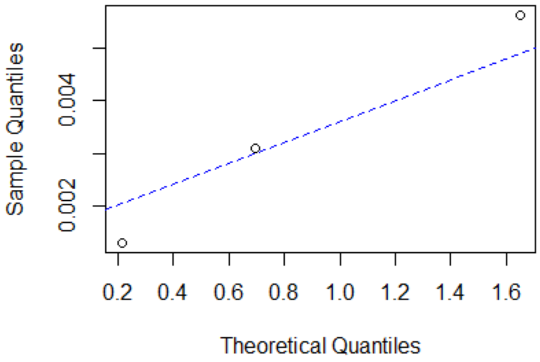


Figure 27: Structure CIF vs. Exponential Q-Q Plot

Graphically, no quantiles of the CIF sample data directly follow the exponential distribution quantile line. Figure 23 especially indicates the poor fit of the theoretical exponential distribution to the sample failure-time data. However, the simplifying effect that the one parameter exponential distribution has on the creation of Markov Chain models must not be understated. Therefore, it is necessary to analyze the impact of the information lost through implementing this simplifying assumption and using the one parameter exponential distribution to represent the failure-time data. The proceeding analyses leverage the Aikake Information Criterion (AIC) metric which estimates the amount of information lost when using a model to represent a sample data set. The following distribution comparisons is merely intended to assess the extent of the time-nonhomogeneous behavior (i.e. how much hazard rate varies dependent upon time) of the failure-time data and the subsequent distribution parameters will not be included in the subsequent Markov chain models.

The AIC separates itself as a useful metric for determining relative goodness of fit as it penalizes the distribution based on the number of free parameters, but rewards a high maximum likelihood value. The AIC does not provide as sense of a model's goodness of fit in the absolute sense, but instead is used to compare candidate models. Its inclusion of a penalty for the number of free parameters discourages overfitting, i.e. the arbitrary increase in model parameters to increase the calculated goodness of fit. The AIC is defined by Aikake (1974) as:

$$AIC = 2k - 2\ln(\hat{L}) \qquad [6]$$

where k is the number of free parameters of the model and $\hat{L}$ is the maximum value of the likelihood function for the model. The smallest AIC value represents the preferred statistical model.

Yet, Claeskens and Hjort (2008) found that in instances of small sample size, where the number of samples is not many times larger than k^2, the use of AIC as the sole model selection metric increases the probability of overfitting. Anticipating this obstacle, Hurvich and Tsai (1989) propose a correction to AIC for instances of finite sample sizes that, in practice, further penalizes a distribution for added free parameters. [7] This corrected AIC, AICc, is defined as,

$$AICc = AIC + \frac{2(k+1)(k+2)}{n-k-2}$$

Where n represents the sample size of the data and not the number of sorties undergone.

Table 5 displays the corrected Aikake Information Criterion (AICc), a relative estimate of the information lost when a model is used to represent a small data set, of two distributions for comparison – the one parameter exponential distribution and the two-parameter Weibull distribution. The one parameter exponential distribution adheres to the Markov property where the hazard rate is constant, while the Weibull is not hampered by this assumption and can therefore represent failure-time behavior that changes over time.

Table 5: Comparison of AICc for Exponential and Weibull Distributions

SubSystem	Exponential AICc	Weibull AICc	Percentage Difference	Preferred Distribution
Electronics	-27.16	-30.81	11.85%	Weibull
Fuel Mgmt.	-26.86	-29.97	10.37%	Weibull
Launcher	-34.55	-40.45	14.59%	Weibull
Operator	-10.37	-16.37	36.65%	Weibull
Propulsion	-21.86	-27.02	19.10%	Weibull
Recovery	-9.58	-18.20	47.39%	Weibull

Structure	-38.21	-49.84	(23.33%)	Exponential

Table 5 shows that the statistical model that best represents six of the seven failure modes is the distribution that allows for variation with respect to time, and the selection of this model has been penalized for the added complication of an additional parameter. According to Table 5, the failure time data for the structure is best represented by the exponential distribution. It is important to note that the number of failure mode observations for this subsystem is much smaller than other subsystems, likely contributing to the advantage of the one parameter distribution.

The preferred statistical model must also consider the application of these CIF data to the overall estimation of reliability. At a higher level of abstraction, the hazard rates of these cumulative incidence functions are applied to Markov chain models that consists of seven failure modes and nine total states. The loss of any information compounds itself as a single subsystem’s failure-time distribution gets simplified. This ultimately impacts the fidelity of the model and the accuracy of the reliability probabilities produced by the Markov chain model.

The use of discrete time non-homogeneous Markov chain models offers numerous benefits over a time homogeneous Markov chain. A non-homogeneous Markov chain model allows for the variation of hazard rate based on a global clock, yet the hazard rate need not change.   Additionally, because discrete time-nonhomogeneous Markov Chains allow for the definition of repair and reliability at each discrete moment in time. This allows future research to examine the impact of changing repair strategy during the system’s operational life. It can be said that the discrete time-nonhomogeneous Markov

Chain modelling technique is the most suitable for the study of this attritable system, based on the fact that the failure-time data for a baseline attritable air vehicle is discrete, the subsystem hazard rates have been shown to vary with respect to time, and that this method affords the greatest flexibility to future reliability and reparability research.

4.3 Reliability Model Results

The determination of the sensitivity of an attritable system's reliability to changes in subsystem reliability and reparability is primarily dependent upon the accurate estimation of the baseline system's reliability. It is shown in the preceding section that the hazard rates of the constituent subsystems can vary with respect to time. Therefore, a Markov Chain model is defined for each respective interval of time, known as a discrete time-nonhomogeneous Markov chain – implemented as a Markov chain list in "markovchain" package (Spedicato, 2015) within R. Figure 28 illustrates the survival function, $\mathrm{S(n)}$, of a list of Markov chains that adhere to the structure of the state-transition diagram in Figure 12 – defined for each time step. Through the calculation of the probability of transitioning to each state for each time step, the system survival function is calculated. The survival function is defined as, $\mathrm{S(n)} = \mathrm{Pr}(\,T > t) = 1 - F(n)$. In this case $F(n)$ represents the probability of entering the absorbing failure state. Figure 28 illustrates the system survival function of the baseline system. This baseline illustration is used as a reference point against which to measure the effect of future reliability and reparability trades.

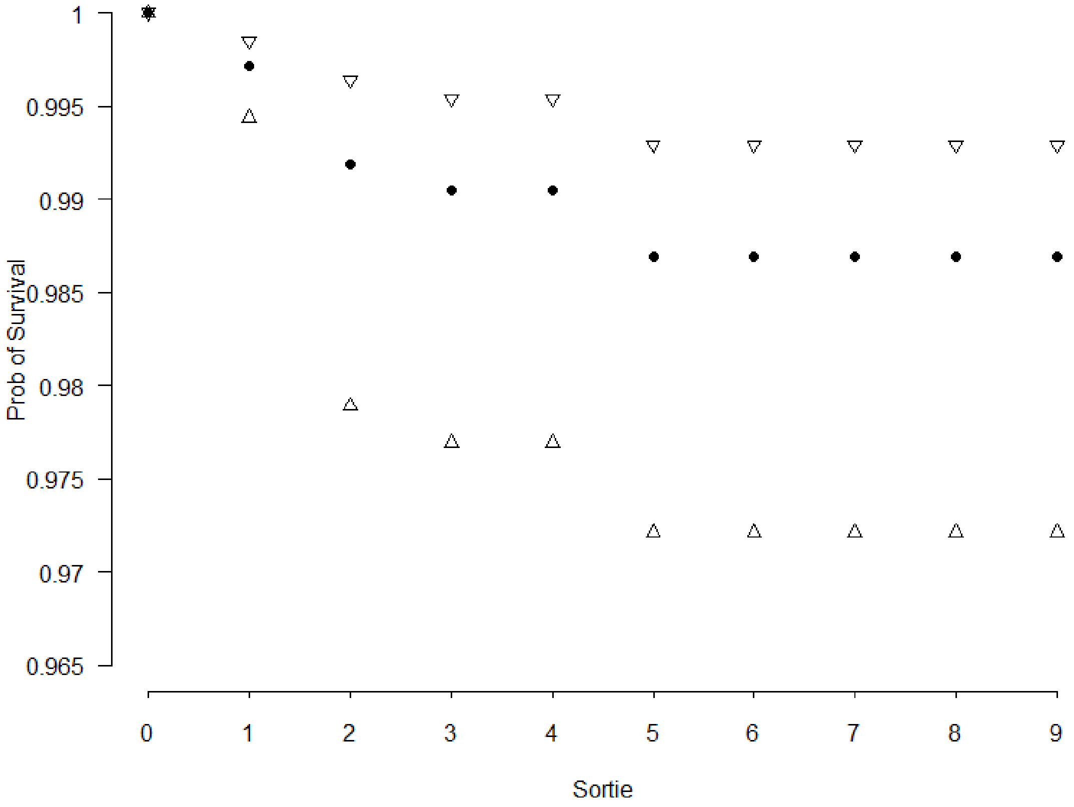


Figure 28: Baseline System Survival Function

In addition to illustrating the probability of unit survival to sortie, n, Figure 28 displays the 95% confidence interval on that probability. These confidence intervals are also defined by the logit transformation presented by Meeker and Escobar that increases the interval's coverage to ensure that it is strictly positive. Future trades in reliability and reparability will vary the discrete time-nonhomogeneous Markov chain model in a manner according to Tables 2 and 3 for reliability and reparability variation, respectively.

4.3.1 Hazard Rate Variation

Figures 29-35 illustrate the effect of varying the hazard rates of a failure mode in a manner consistent with Table 2. The hazard rates are varied by a fixed percentage as it applies a consistent variation of the hazard rate for all nine time intervals defined in the discrete time-nonhomogeneous Markov chain models. Based on discussion in previous chapters, these models only estimate reliability up to the ninth sortie due to lack of data for systems that have undergone ten or more sorties. The direct specification of hazard rate, as opposed to other reliability metrics like mean time between failure (MTBF), avoids the pitfalls of simplifying the failure-time data behavior to adhere to the constant hazard rate assumption. The impact of this assumption on the fidelity of the reliability estimation is addressed in the preceding section.

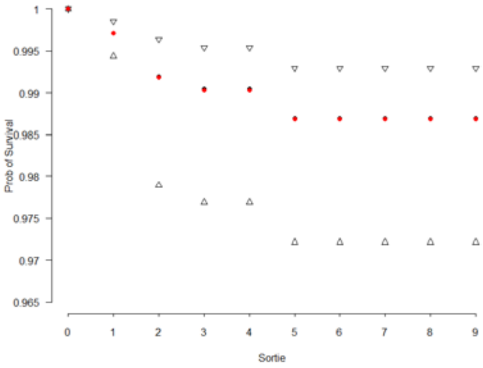


Figure 30: S(n) with Altered Electronics Subsystem

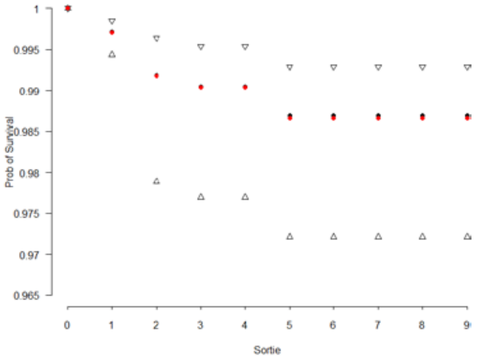


Figure 29: S(n) with Altered Fuel Mgmt Subsystem

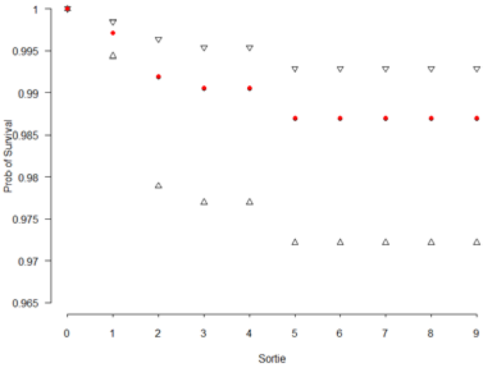


Figure 32: S(n) with Altered Launcher Subsystem

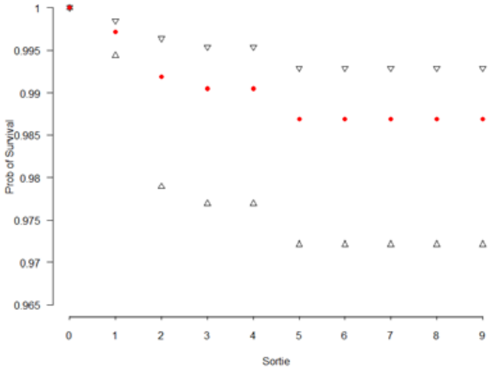


Figure 31: S(n) with Altered Operator

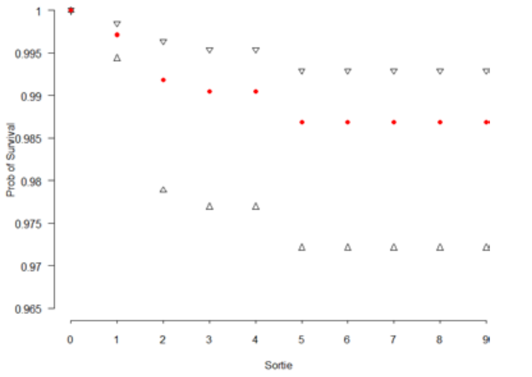


Figure 34: S(n) with Altered Propulsion Subsystem

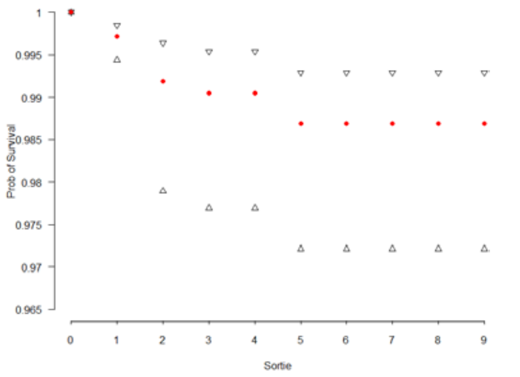


Figure 33: S(n) with Altered Recovery Subsystem

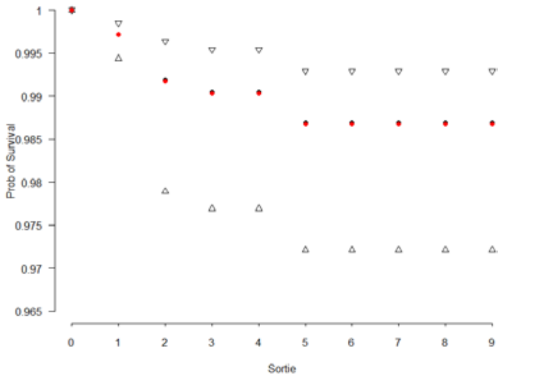


Figure 35: S(n) with Altered Structure

These figures illustrate that the trades outlined in Table 2 do not represent a significant impact on overall system survivability. Their exact impact is illustrated in proceeding section, but it important to note that the survival function of a modified system, altered such that one of its subsystems fails at an increased hazard rate, correlates to a decreased probability of survival to the next interval in time. Conversely, a system that consists of an improved – i.e. a lower hazard rate – subsystem has an increased probability of survival to the next sortie. Figures 29-35 illustrate this intuitive result but are critical to the quantification of the sensitivity of the system survival function to changes in subsystem hazard rate. Figure 36 illustrates this sensitivity by quantifying the percentage change in $S(n)$ for a given change in subsystem hazard rate.

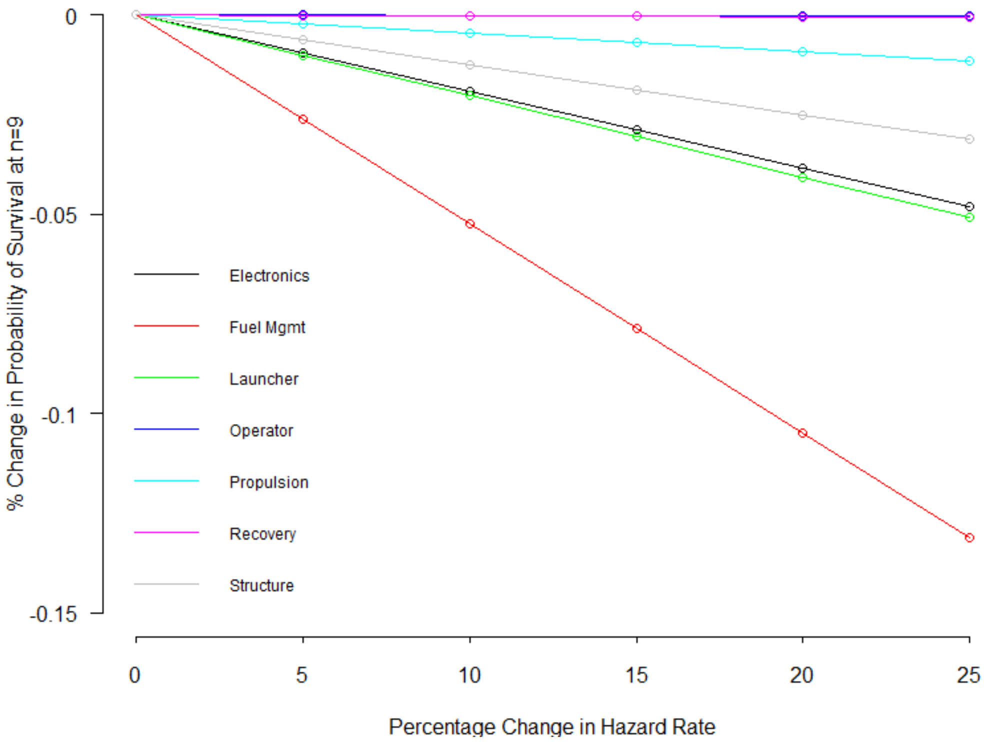


Figure 36: Sensitivity of Trading Subsystem Hazard Rate on Survival Function

Figure 36 illustrates that the system survival function is most sensitive to fixed changes in hazard rate of the subsystems that have the highest hazard rates. This is illustrated by the fact that the system survival function most sensitive to the three subsystems with the highest hazard rate – i.e. the fuel management, launcher, and electronics subsystem respectively. However, this illustration of system survival function sensitivity demonstrates interesting characteristics for those subsystems with much lower sensitivities. For example, Figure 36 shows that the system survival function is more sensitive to changes in the hazard rate of the structure than the propulsion subsystem. This is likely because of the fact that system survival is dependent on two transition processes as illustrated in Figure 12. These two processes are the transition of the system from an operational state to a failed, but reparable, state in addition to the transition from the failed state to the absorbing failure state, or the destroyed state. The high probability of transitioning to the destroyed state after a structural failure makes $S(n)$ more sensitive to increases in the hazard rate of the structure. Conversely, the high probability of repair for a failed propulsion subsystem to be repaired, at least of the baseline system, decreases the sensitivity of $S(n)$ to reliability trades for these subsystems.

This research only examines the sensitivity of the system to trades in subsystem failure probability as these transitions most clearly represent the reliability of a system. The transitional probability of stepping between a failed state to the destroyed state represents a failure's consequences. Trades of these probabilities do not trade on subsystem reliability, but instead trade on the strategies to mitigate the negative consequences of a failure mode.

4.3.2 Reparability Variation

As this research seeks to determine the impact of altering reparability on overall system reliability, Figures 37 ad 38 illustrate the marked impact that specifying Propulsion subsystem or Structural failures irreparable has on the system survival function. Trading the reparability of these subsystems investigates the utility of attritable maintenance decisions under consideration by researchers. These figures illustrate the impact of prohibiting the repair of a critical subsystem, such as the propulsion subsystem and structure, has on the system-level probability of survival. Just as in previous figures, the baseline cost risk is presented with 95% confidence intervals based off of the upper and lower estimates of Meeker and Escobar's transformation for the effect of small sample sizes.

Figure 37 illustrates the noteworthy impact that trading reparability can have for a subsystem with comparatively high hazard rates. It shows that the probability of system survival is much less than the estimate and outside of the confidence intervals within three sorties.

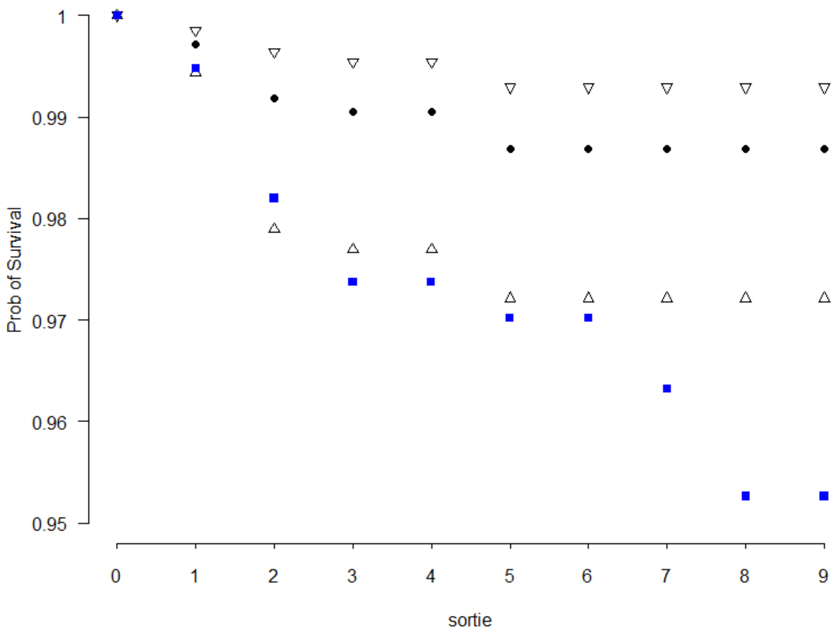


Figure 37: System Survival Function without Propulsion Subsystem Repair

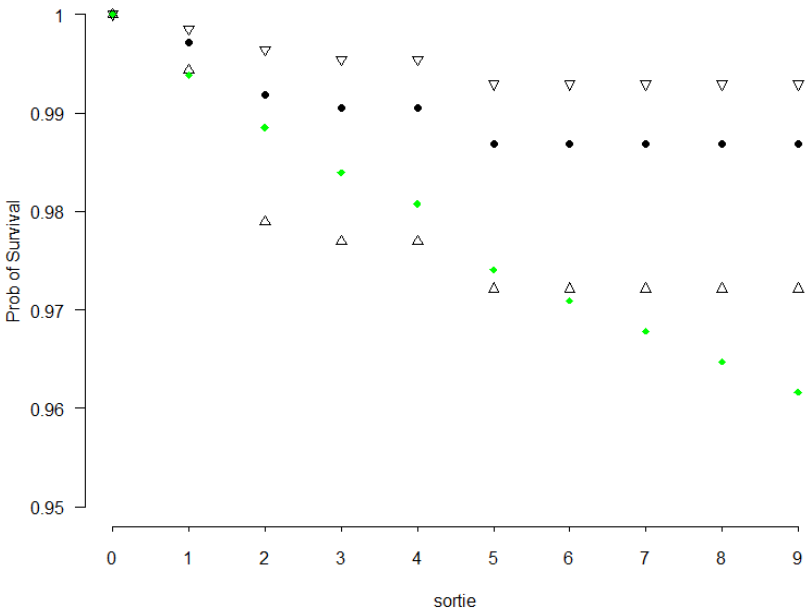


Figure 38: System Survival Function without Structural Repair

However, Figure 38 illustrates the minor impact of trading reparability for a system whose hazard rate is lower. While there is an estimated decrease in the survival function as time goes on, there is no significant decrease in the probability of survival until the third sortie. These figures illustrate the spectrum of effects that trading subsystem reparability has on system-level survival. Trading reparability for a subsystem with a high probability of failure significantly decreases probability of survival. Meanwhile, trading reparability for a subsystem with a lower hazard rate decreases the impact on system probability of survival; yet, the probability of survival decreases all the same.

Lastly, the elimination of a repair action also impacts the estimated cost risk of each sortie as a non-repairable subsystem incurs no cost of repair. Though the system's sensitivity to changes in subsystem reliability and reparability are quantified, the impacts of these trades on system costs are as yet unknown. The impact of a changing system survival function on system cost risk is addressed in proceeding sections.

4.4 Cost Risk Estimation

Estimating the impact of subsystem reliability and reparability for the possibility of lower cost risk is perhaps the ultimate objective of the Low Cost Attritable Aircraft Technology (LCAAT) demonstration program. This impact could affect the implementation of an attritable design alternatives or maintenance decisions as they must demonstrate an acquisition, sustainment, or operational advantage. As previously discussed, the calculation of system cost risk uses the state probabilities, an output from the aforementioned discrete time-nonhomogeneous Markov chain models, as well as the example cost parameters outlined in Table 4. Thus, the cost risk of each sortie is defined by Equation 3.

Figure 39 illustrates the baseline attritable air vehicle's cost risk for each sortie. Note that a system's first sortie cost risk is relatively low, consisting mostly of the regeneration cost of the vehicle. However, as a system undergoes additional sorties the probability of failure and subsequent transition to the absorbing failure state, increases. This phenomenon of an increased probability of destruction is responsible for the general upward trend of the absolute cost risk curve.

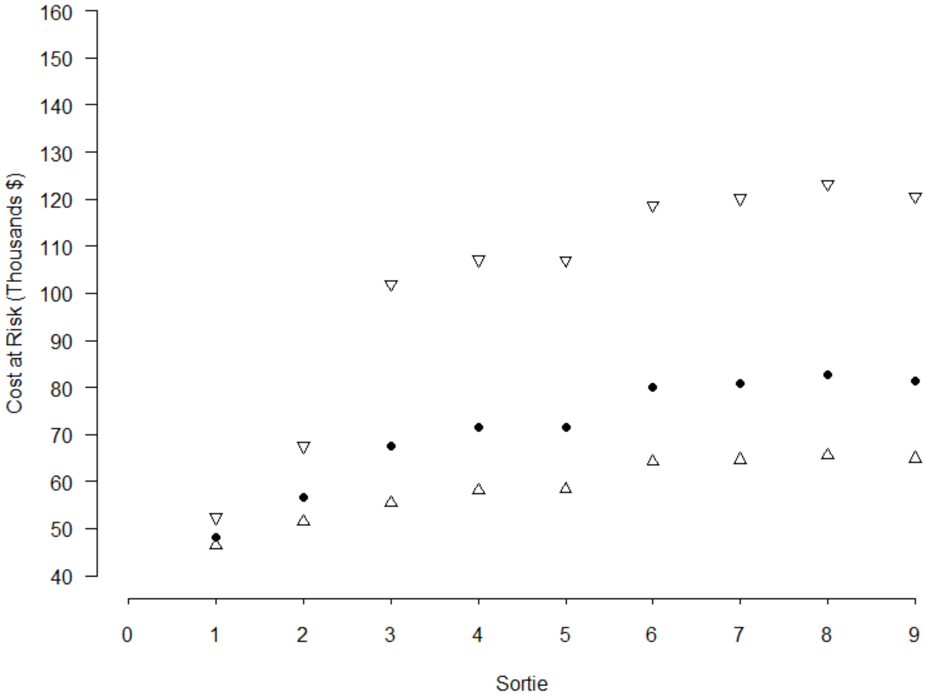


Figure 39: Absolute System Cost Risk of the Baseline Attritable Air Vehicle

Yet, there are intermittent decreases in the absolute cost at risk, for example between the fourth and fifth sorties of a system. Here, the effect of the various costs to repair a subsystem are illustrated. Due to specification of hazard rate for each sortie individually, the noticeable decrease in the number of failures observed for the fuel management, launcher, operator, and recovery subsystems between the fourth and fifth sortie decreases the estimated cost risk. On the absolute scale, this decreased cost risk is on the order of a one percent change and does not significantly shift the bounds of the confidence interval.

Figures 40 and 41 present an estimation of cost risk for the cases where the reparability of the propulsion subsystem or structure is traded, respectively. The justification for these research excursions are presented in Table 3. These figures differentiate themselves from the calculation of system survival probability, as presented in Figure 37 and 38, as the repair cost of the irreparable subsystem is decreased to zero. Therefore, the cost of these subsystems' repair does not contribute to the cost at risk for a given sortie. Still, the probability of transitioning into the absorbing failure state, $F(n)$, increases due to the inability to repair the subsystem.

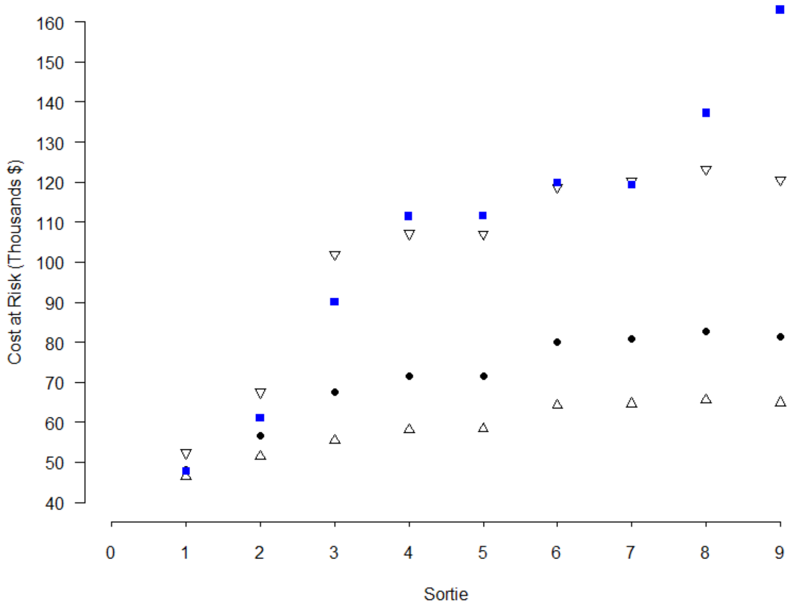


Figure 40: Absolute Cost Risk per Sortie for Non-Repairable Propulsion Subsystem

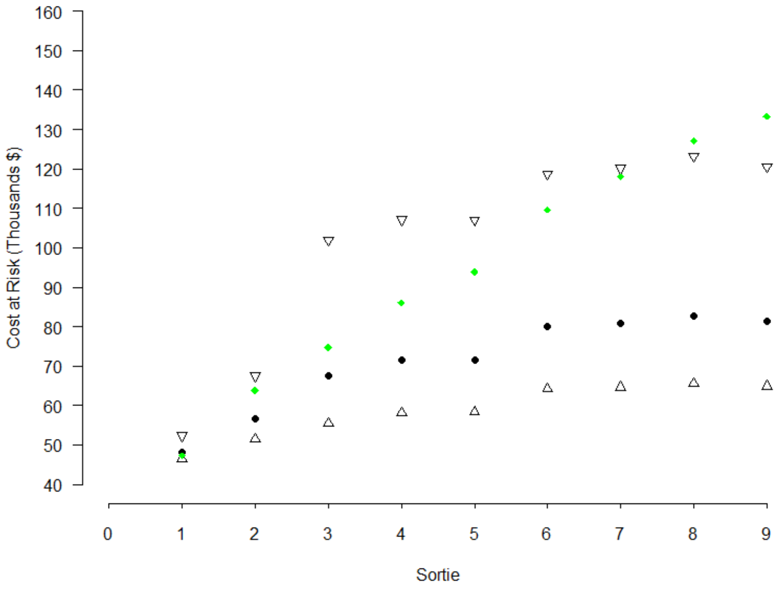


Figure 41: Absolute System Cost Risk per Sortie for a Non-Repairable Structure

This increase in $F(n)$ increases the cost risk as the increase in $F(n)$ is compounded by the greatest cost category, AUFC. This cost category is an order of magnitude greater than the cost to repair or replace a subsystem as it is a collection of these other cost categories. Therefore, any increases in $F(n)$ have a significant effect on cost at risk. Figures 40 and 41 demonstrate the spectrum of consequences that prohibiting subsystem repair has on cost risk. The propulsion subsystem's hazard rate is greater than the structure's hazard rate and Figure 40 shows that prohibition of propulsion repair has the expected effect of significantly increasing cost risk – nearly doubling the estimated cost risk by the third sortie. Conversely, the structural subsystem experiences the lowest hazard rate of all seven subsystems. Yet, Figure 41 illustrates that even considering the nullification of structural subsystem repair costs, the decision to prohibit structural repair still increases the estimated system cost risk.

Calculating the effect that trading reparability on system-level cost risk is basic when compared to the comparison of trading hazard rates for multiple design alternatives. As discussed in preceding sections, the hazard rates of various subsystem alternatives could illustrate hazard rates that differ from the baseline system. The subsystem acquisition cost of these alternatives may also deviate from the baseline subsystem cost. These subsystem acquisition cost variations not only impact subsystem repair costs – as this is function of the repair-to-buy cost ratio – but also affect the average unit flyaway cost (AUFC). The aggregation of these effects make it difficult to determine whether a design alternative has a positive impact on cost risk and AUFC.

This research assesses the impact that these design alternatives have on cost risk for two subsystem alternatives, the electronics subsystem and propulsion subsystem. As discussed in Table 2, the effect of using commercially available unmanned aerial vehicle flight control systems that do not adhere to MIL-SPEC standards – and thus trade attributes like redundancy for decreased cost – is unknown. Similarly, a baseline system modified to accept a commercial-off-the-shelf propulsion system may have a different hazard rate as well as a different subsystem acquisition cost. The comparative abundance of design alternatives that meet the requirements for these subsystems lends itself to further investigation into which design alternatives present an advantage in acquisition cost as well as cost risk.

To simplify the identification of advantageous design trades the percentage decrease in subsystem acquisition cost necessary for a given change in hazard rate to place the equivalent amount of cost at is calculated. This is referred to as the equivalent cost risk line in Figures 42 and 43. A design alternative that falls on this equivalent cost risk line places the same amount of cost at risk as the baseline air vehicle, but could represent a system with a lower AUFC. Additionally, the range of percentage change in hazard rate evaluated for these two subsystems is dependent upon the specific design alternatives under investigation.

The range of electronics subsystem hazard rate variation under investigation estimates the possible decrease in reliability realized by changing physical attributes such as redundancy to decrease cost. For reference, a five percent increase in hazard rate equates to a five percent decrease in Mean Time Between Failure if the related failure-time data were assumed to have a constant hazard rate.

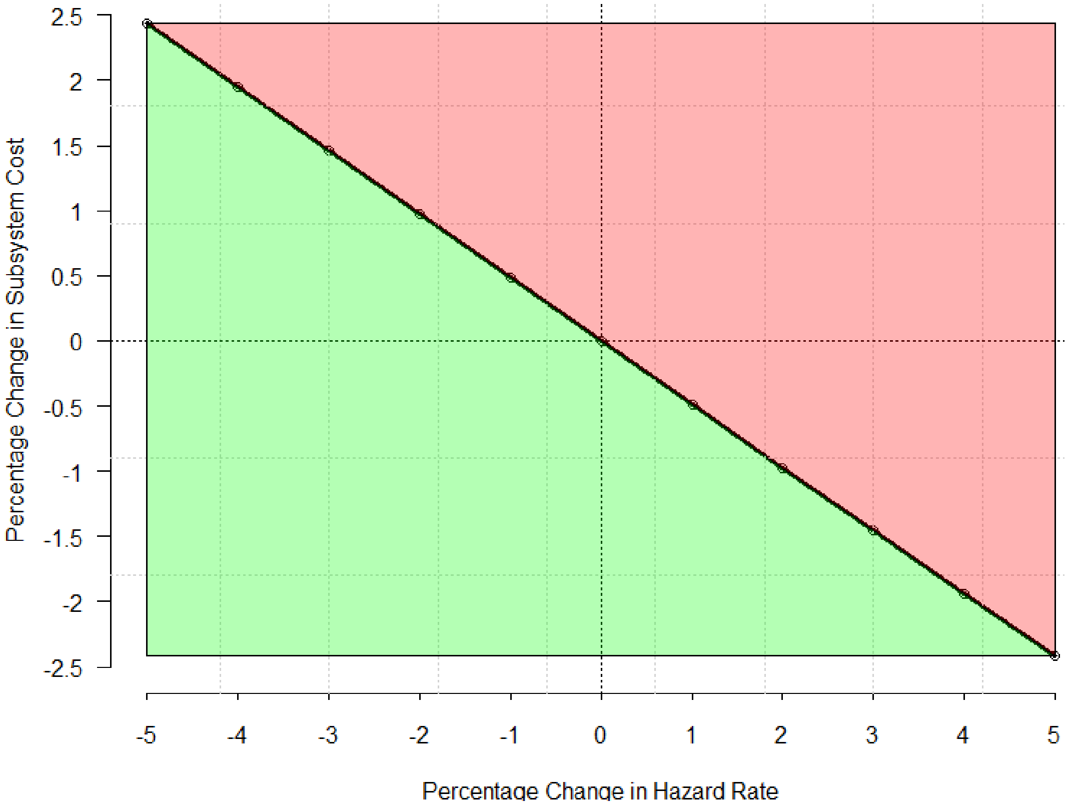


Figure 42: Equivalent Cost Risk for a Trade in Electronics Hazard Rate

If a design alternative that falls within the region below the equivalent cost risk line where the percentage decrease in subsystem acquisition cost is greater than the percentage increase in hazard rate that design alternative trades reliability for a lower AUFC and cost risk in an advantageous manner. The region of beneficial trade space is shown in the green regions for Figures 42 and 43. Conversely, a design alternative that falls on the other side of this equivalent cost risk curve represents a reliability trade that does not decrease cost risk. Note that there is a region of this trade space that shows a decrease in subsystem acquisition cost, yet an increase in cost risk. This disadvantageous region of trade space is represented by the color red.

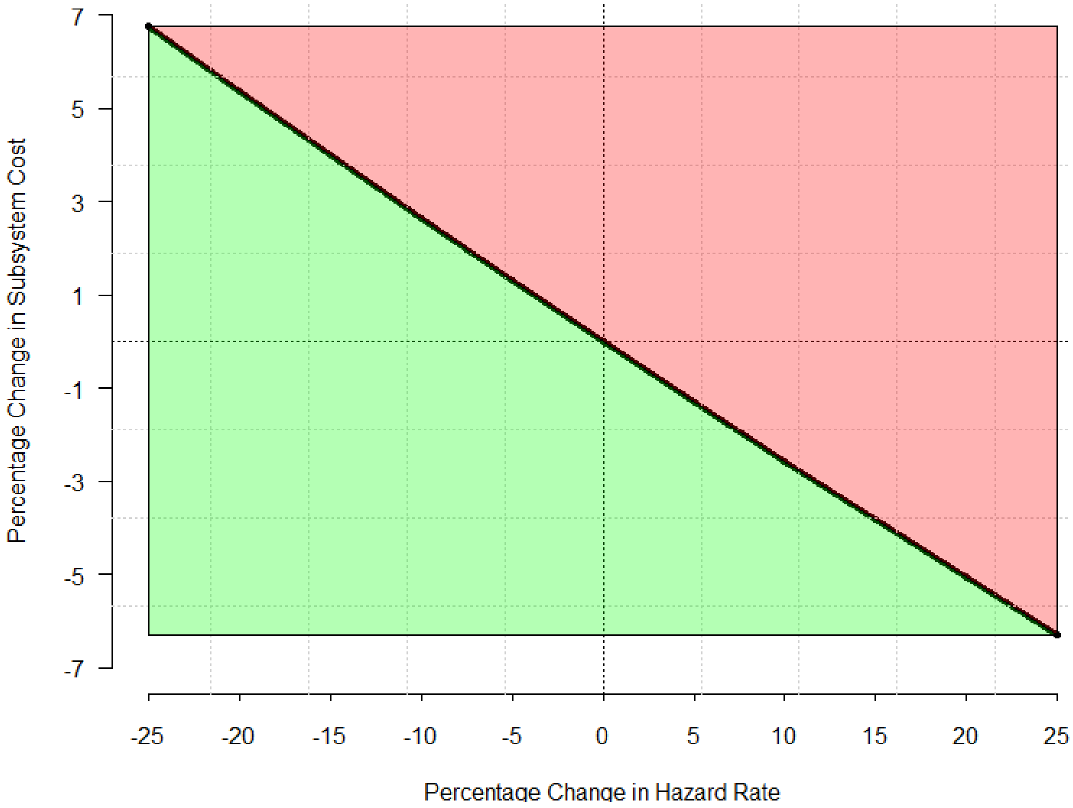


Figure 43: Equivalent Cost Risk for a Trade in Propulsion Hazard Rate

Figure 43 illustrates the equivalent cost risk trade space for the propulsion subsystem. The range under investigation is much wider than for the electronics subsystem as there are many existing design alternatives that report a wide spectrum of maintenance and failure intervals (many of them reported in MTBF). Analysis of the equivalent cost risk line shows that the slopes of the equivalent cost risk lines of Figures 42 and 43 differ. The equivalent cost risk line of Figure 43 is much shallower than the equivalent cost risk line of Figure 42, even though Figure 43 displays a much wider range of variation. This implies that for a given change in propulsion subsystem hazard rate the accompanying necessary percentage change in subsystem acquisition cost is much less than is required for a trade of electronics subsystem hazard rate. Thus, there exists a greater opportunity to make a trade that both decreases cost risk as well as AUFC for the propulsion subsystem. These illustrations of equivalent cost risks are not to intended to prioritize one attritable subsystem trade over another, as this requires the consideration of many other factors, but instead inform analysts about the point at which a design trade becomes advantageous in terms of cost risk.

4.5 Conclusions

The method outlined in the proceeding sections uses the hazard rate of the cumulative incidence function to create discrete time-nonhomogeneous Markov chains to estimate both the system survival function as well as the cost risk of the system. Preceding sections also outline the suitability of this method for the analysis of fielded attritable system failure-time data as it provides the most flexibility in the specification of system reliability and reparability. Analysis of the cumulative incidence function is shown to be the most appropriate for the study of fielded baseline attritable system failure-time with competing risks and a unique censoring scheme. Also, the discrete time-nonhomogeneous Markov chain method is shown to more accurately represent the failure-time data, in addition to providing the flexibility to trade reliability and reparability through the creation of multiple alternate Markov chain models. Finally, the calculation of cost risk is both a metric that is grounded in the literature as well as a true estimation of the consequence of a trade in reliability and reparability for the operator.

Trades in attritable air vehicle subsystem reliability are performed by varying subsystem hazard rates and calculating the resultant percentage change in the probability of system survival, as illustrated in Figure 36. This figure shows that variations in hazard rate for a subsystem with high probability of failure in the next small interval of time have the greatest impact on system survival $S(n)$. This figure also demonstrates that system survival is impacted not only by a subsystem's propensity to fail, but also by the risk of destruction presented by that failure mode. The consequences of trading subsystem reparability are also presented in Figures 37 and 38. It is clear that regardless of the magnitude of the subsystem's hazard rate, the decision to trade the ability to restore it to an operational state will always decrease the probability of survival to the next sortie. The magnitude of this effect is dependent only upon the probability of failure, as this trade specifies that a system will always be considered destroyed given a failure of this subsystem.

Next, the impact of trading reliability and reparability on system cost risk is examined, using example costs defined in Table 4. When trading the ability to repair a system the estimated cost risk increases in every case. Despite the nullification of the cost to repair the system, the increased probability of transition to an unrecoverable failed state always increases the system cost risk. The transition to this unrecoverable state has the greatest impact on the calculation of cost risk as the average unit flyaway cost is an order of magnitude greater than the subsystem acquisition cost or its average cost to repair.

Finally, the impact of trading subsystem hazard rate on cost risk is examined through the study of the electronics and propulsion subsystems. These subsystems are examined as there is a high likelihood that technically acceptable commercially available design alternatives exist for these subsystems. Figures 42 and 43 present the change in cost necessary for a given trade in reliability to for a cost risk equivalent to the baseline design. The determination of the necessary change in cost for a given trade in reliability allows designers to assess design alternatives based on their overall impact on cost and not just on reliability.

The determination of the change in subsystem acquisition cost necessary for an equivalent cost risk also illustrates the large design space for trades in reliability (i.e. increases in hazard rate) and that some trades that decrease cost risk could also decrease average unit flyaway cost. Design alternatives that fall into the advantageous trade space, yet accept a higher subsystem acquisition cost for the accompanying increase in reliability are trades that many designers are comfortable with making. These trades, as well as the trades that both increase reliability and decrease subsystem acquisition cost, are the trades that are often implemented on survivable system meant for long life. This “Survivable and Maintainable” trade space falls into the second quadrant of Figure 44, a representative equivalent cost risk graph.

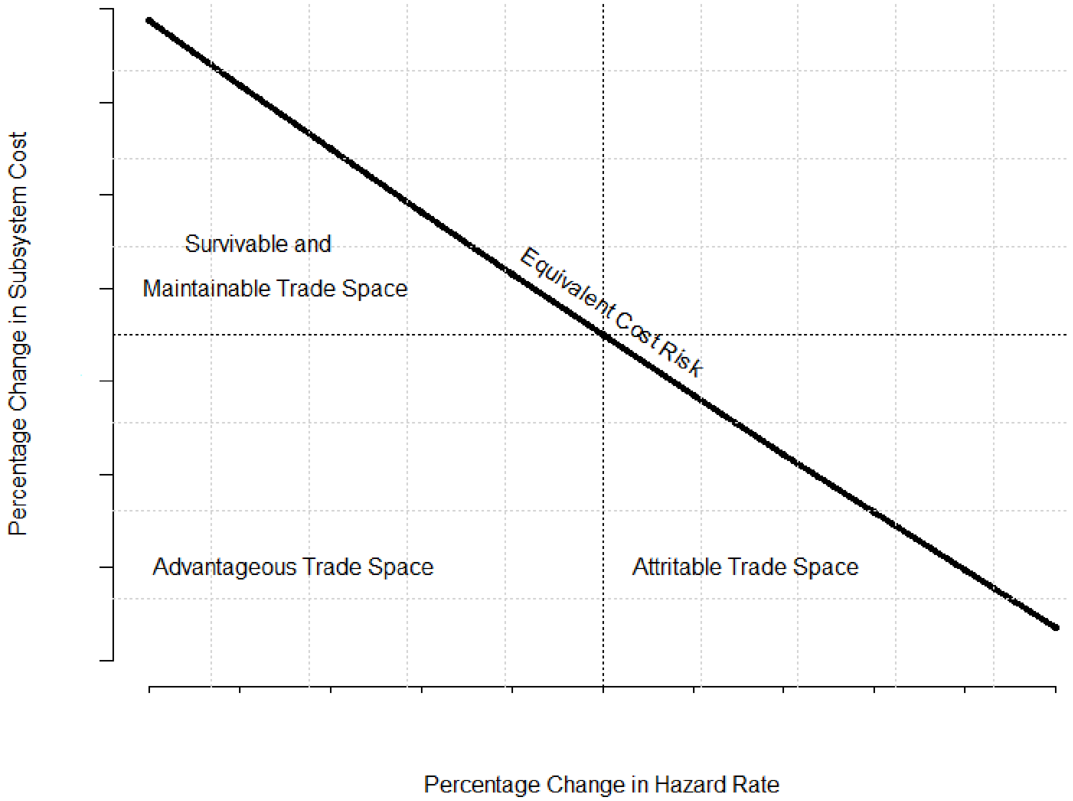


Figure 44: Attritable and Survival Design Spaces based on Equivalent Cost Risk

However, there exists an alternative design space that both decreases the system cost risk and the subsystem acquisition cost, yet allows for decreased reliability for reparable systems. This attritable design space exists in the fourth quadrant of Figure 44, where subsystem hazard rate increases yet the subsystem acquisition cost and the system cost risk decreases from that of the baseline system. The identification of this trade space for a similar air vehicle meets AFRL's original objective to trade system attributes like reliability and reparability to achieve a decrease in system costs for a system "whereby virtue of its cost, the loss of the aircraft could be tolerated" (AFRL/RQKP, 2015, p.1).

5.0 Conclusions and Recommendations

5.1 Chapter Overview

The original objective of this research was to express the impact of trading subsystem reliability and reparability on system-level reliability and overall cost risk. This chapter will review whether the selected research methodology conveys these effects by reviewing the investigative questions, ensuring that this research meets its objective. It also discusses the opportunities to extend this method and use the data gathered on fielded attritable air vehicles for future efforts. The chapter concludes with a discussion of the research's significance and outlines how it can be employed to inform decision-makers about the value of making trades in the attritable design space.

5.2 Investigative Question Review

To express the impact of trading subsystem reliability and reparability in order to realize a decrease in system cost, this research defined three primary research questions. First, which metrics and methods are suitable for the estimation of reliability and costs for attritable systems? Second, how sensitive is an attritable system's reliability to changes in subsystem reliability and reparability? Finally, what effect does varying subsystem reliability and reparability have on the cost at risk of an attritable system? Ultimately, the advantageous design space that trades system attributes like reliability and reparability is identified by answering these three investigative questions.

5.2.1 Metrics and Methods Suitable for Attritable System Reliability

This research discussed the abundance of reliability metrics and modelling techniques intended to describe the reliability and maintainability of a system. Two overarching modelling techniques were identified: combinatorial and state-space techniques. It was determined that the requirement for repair of the vision attritable air vehicle made combinatorial techniques, such as Reliability Block Diagrams and Fault Tree Analyses, unable to model the reliability of such as system. The ability to model dynamic systems made state-space reliability modelling techniques appropriate to apply to the study of attritable air vehicles. However, it was revealed that state-space modelling techniques like Markov chains also suffer from an inability to represent reparable systems that change over time, complicating the modelling of this system.

It was also found that legacy reliability metrics often require the application of simplifying assumptions which conceal the underlying behavior of the failure-time data they are meant to represent. As is discussed in Section 4.2, the amount of information lost by making these assumptions are compounded when these statistics are applied to reliability models like Markov chains. This research overcame these limitations through the employment of discrete time-nonhomogeneous Markov chains that vary hazard rate directly by defining the probabilities of transitioning between states for each interval in time.

In addition to employing this flexible modelling technique, this research also found that the calculation of each failure mode's cumulative incidence function was the most appropriate way overcome the presence of competing risks. A competing risk, or competing failure mode is a failure mode that competes for observation against all other failure modes. The calculation of each subsystem's hazard rate of the discrete cumulative incidence function is used as an input into this research's discrete time-nonhomogeneous Markov chain models.

5.2.2 Sensitivities to Variations of Reliability and Reparability

Once the suitability of the competing risks analysis input into a discrete time-nonhomogeneous Markov chain method was established, the model was used to calculate baseline system sensitivities. Figure 36 illustrated the maximum percentage change in $S(n)$, that is the probability of system survival to that sortie, for a given trade in hazard rate. In addition to quantifying the expected percentage change in $S(n)$, this figure also ranks the subsystem sensitivities relative to each other. It was found that the probability of system survival is most sensitive to trades in the hazard rates of the fuel management, launcher, electronics, structure, propulsion, recovery, and operator subsystems, respectively. It was also discussed how these sensitivities differed from a simple calculation of the order of magnitude of hazard rate due to the fact that $S(n)$ is affected by both the propensity of the subsystem to fail, but also the probability of the failure mode to transition to an absorbing failure state.

The second objective of this research's quest to explore reliability and reparability impacts was fulfilled through the creation of discrete time-nonhomogeneous Markov chain models that prohibit the repair of the propulsion and structural subsystems. These models negate subsystem reparability, ensuring that the system will always transition to the absorbing failure state after a failure of either of these subsystems. Figures 37 and 38 illustrate the resultant decrease in the system survival function when the reparability of these subsystems are traded. These subsystems were selected to examine further as these are actual trades under consideration by AFRL researchers and they illustrate the spectrum of impact on $S(n)$ as the propulsion system has a relatively high hazard rate while the system's structure has a comparatively low hazard rate.

Figure 37 illustrated that the probability of system survival to the next sortie is markedly decreased when the ability to repair the propulsion subsystem is traded. Figure 38 illustrates a similar result, but the effect is less pronounced. This slight decrease in system probability of survival is due to the fact that the structure experiences a hazard rate that is the lowest out of all seven subsystems. Thus, it was determined that the probability of system survival to the sortie always decreases when subsystem reparability is prohibited.

5.2.3 Consequences of Trading Reliability and Reparability on Cost Risk

In the case of an attritable system, trades in subsystem reliability and reparability are ultimately made to hopefully realize some level of savings with regard to system costs. This research accounts for these system level costs through the use of the metric of cost risk as well as average unit flyaway cost. Cost risk is determined by applying the example costs identified in Table 4 to the equation defined in Section 3.4. The baseline cost risk of the system is illustrated in Figure 39 while Figures 40 and 41 show the impact of trading reparability of the propulsion and structural subsystems, respectively. They show that even though the repair costs for these subsystems are nullified, the estimated cost risk increases.

The estimation of cost risk for design alternatives is much more complicated as existing subsystem alternatives differ in both subsystem reliability and subsystem acquisition cost. Therefore, Figures 42 and 43 illustrate the necessary percentage change in subsystem acquisition cost for a given trade in subsystem hazard rate to place the equivalent amount of cost at risk. Design alternatives that demonstrate cost and reliability properties that are below and to the left of the equivalent cost curve present an advantageous trade in cost risk over the baseline design. Conversely, alternative subsystems that are that are marginally more reliable, yet much more expensive are designs that increase the system cost risk over the baseline. Design trades that are not advantageous fall into the red region of Figures 42 or 43.

Furthermore, Figure 44 differentiates between two design spaces shown within the advantageous cost risk design space: those for survivable systems and those for attritable systems. Design trades that allow for increased subsystem acquisition cost, a subsequently average unit flyaway cost, to realize an increase in system reliability represent one of the classic trades made for survivable systems. In these cases, designers have determined that an increase in reliability is worth the increased acquisition cost. Yet, Figure 44 also illustrates that there is a design space that allows for a decrease in reliability for lower subsystem cost, i.e. the attritable design space, as long as the design alternative falls below and to the left of the equivalent cost risk line. This research allows analysts to identify the boundaries of this equivalent cost risk line for the subsystems mentioned, thereby allowing them to make these attritable trades.

5.3 Recommendations for Future Research

The method identified in this research can be further refined or expanded for use as an analysis tool for attritable system analysts. One such application is to quantify the impacts of decisions under consideration by system designers and maintainers. The discrete time-nonhomogeneous Markov chain technique allows future research to define repair or failure-rates at each respective interval in time. This fidelity allows maintainers to quantify the impacts of altering reparability during the operational life on the probability of survival function as well as cost. Additionally, this flexibility to define failure rate at each interval of time also opens the door to examining the impact of decreasing acceptance and non-destructive testing may have on infant mortality and cost risk.

This research, which characterizes system-level impacts of varying subsystem reliability and reparability, could be expanded for use in fleet-level studies. These fleet-level studies, which would require estimates of usage and loss rate due to external circumstances, could be used to determine the impact of trading reliability and reparability on a fleet. The system survival probabilities, specified by this research, could be used as an input to fleet-level queueing models to determine fleet size requirements. Additionally, a queueing model could be built to simulate the effect of fleet-size maintenance decisions. Researchers have sought a way to characterize the effect that cannibalization – that is, the reuse of serviceable components taken from a decommissioned, or destroyed, system – can have on fleet maintenance and sparing costs for such an attritable system. Future research in this area could further decrease the cost risk of attritable air vehicles operations.

Further research can also be conducted on the operational effects of developing and fielding a system whose reliability has been traded for a lower system cost. The operational effect of employing fleets of attritable air vehicles must be determined and presented to Air Force strategic planners. The probabilities of failure, specified in this research, could be used as inputs into both engagement and theatre-level campaign models to determine the effect that trading system reliability and reparability will have on the number of attritable air vehicles that need to be launched to achieve a prescribed mission. Additionally, polls of unmanned air vehicle system operators and customers of their capabilities can be taken to determine the lower thresholds of availability that their missions require. Such assessments would ease the validation of attritable air vehicle fleet suitability and effectiveness.

Beyond the operational consequences of trading reliability and maintainability, further research on methods to incentivize designers and maintainers to follow the "attritable" design philosophy is important in the implementation of attritable weapon system programs. Many system designers and maintainers admit their aversion to trading key system attributes like reliability and reparability simply to achieve a lower cost. In the case of the baseline air vehicle described throughout this research, the system is classified by the DoD as a "contractor supported weapon system." This classification places the responsibility of maintaining pre-prescribed reliability, maintainability, and availability (RAM) metrics solely on the contractor with no motivation to decrease per-unit or maintenance costs. As the system is fielded, there is a disincentive for maintenance contractors to trade these RAM metrics for lower costs for fear of reneging on their contract. The execution of an attritable weapons system program must take a holistic approach to incentivizing this new design philosophy to effectively achieve perceptible decreases in cost risk.

5.4 Significance of Research

The preceding conclusions of this research advances AFRL's quest to trade the system attributes of reliability and reparability to achieve a such a low cost "whereby virtue of its cost, the loss of the aircraft could be tolerated" (AFRL/RQKP, 2015, p.1). The modelling technique demonstrated by this research provides the most flexibility to analyze these trades' effect on the system as it can model a dynamic system that changes with time. Analysis of these models illustrate the cost of these trades in both system probability of survival in addition to cost risk. By quantifying the effects of these trades, decision makers can use this research to guide attritable system development. Attritable systems leveraging the findings of this research will ultimately meet the intent of the Third Offset Strategy by delivering a necessary capability at a lower average unit flyaway cost, thereby overcoming a numeric disadvantage in the future.

Bibliography

AFRL/RQKP. (2015). *Low cost attritable strike unmanned aerial system demonstration – notice of contract action (NOCA).* BAA-AFRL-RQKP-2015-0004. Retrieved on July 20, 2016, from https://www.fbo.gov/index?s=opportunity&mode=form&id=14b63e83b3345e3bc8fdc5d02d288f62&tab=core&_cview=1

Barnard, R. W. A. (2008). What is wrong with reliability engineering?. In *INCOSE International Symposium* (Vol. 18, No. 1, pp. 357-365).

Blanchard, B. S. & Fabrycky, W. J. (1998). *Systems Engineering and Analysis, 3rd Ed.*, Prentice Hall

Boyd M. A. (1998). An introduction to Markov modelling concepts and uses. In tutorial at the *Reliability and Maintainability Symposium*. 19-22 January 1998, Anaheim, California.

Butler, R. W., & Johnson, S. C. (1995). *Techniques for modeling the reliability of fault-tolerant systems with the Markov state-space approach.* (NASA-RP-1348). Hampton, VA: NASA Langley Research Center.

Chew, S.P., Dunnett, S.J., & Andrews, J.D., (2007). Phased mission modeling of systems with maintenance free operating periods using simulated Petri-nets. *Proceedings of the 17th Advances in Risk and Reliability Technology Symposium*, Longborough, UK, April, pp. 219-230.

Claeskens, G., & Hjort, N. L. (2008). *Model Selection and Model Averaging.* Cambridge University Press

Cost Assessment and Program Evaluation. (2015). *Department of Defense Instruction 5000.73: cost analysis guidance and procedures.* Washington D.C.: Office of Director, Cost Assessment and Program Evaluation.

Defense Acquisition University (2010). *Cost Estimation Methodologies*. ACQuipedia. Retrieved on 27 Semptember 2016 from, https://dap.dau.mil/acquipedia/Pages/ArticleDetails.aspx?aid=8e8f5bf3-f517-4e98-8c07-80bb8670a830

Department of Defense. (2015, Jan 7). *Operation of the defense acquisition system.* (DOD Instruction 5000.02). Washington, DC: Undersecretary of Defense for Acquisition Technology and Logistics.

Department of Defense Standard Practice (MIL-STD). (2011) *MIL-STD-881C: Work breakdown structures for defense materiel items*. Washington, DC.

Director, Operational Test and Evaluation. (1999). RQ1-A predator unmanned aerial vehicle. *FY99 Annual Report*. Retrieved on 12 August, 2016, from http://www.dote.osd.mil/pub/reports/FY1999/pdf/af/99predator.pdf

Freels, J. K. (2013). *Modelling reliability growth in accelerated stress testing*. (doctoral dissertation), Air Force Institute of Technology. Wright-Patterson AFB, OH, Graduate School of Engineering and Management.

Gray R.J. (2014). Package '*cmprsk*'. Subdistribution analysis of competing risks. R package version 2.2–7.

Gray, R. J. (1988). A class of K-Sample tests for comparing the cumulative incidence of a competing risk. *Annals of Statistics*, vol. 16: 1141-1154.

Gooley, T.A., Leisenring, W., Crowley, J., & Storer, B.E. (1999). Estimation of failure probabilities in the presence of competing risks: new representations of old estimators. *Statistics in medicine*, vol. 18.6, p. 695-706.

Headquarters, United States Air Force (HAF). (2014). *United states air force RPA vector: vision and enabling concepts 2013-2038* (U). Washington D.C.: Headquarters, United States Air Force.

Headquarters, United States Air Force (HAF). (2015). *Air force future operating concept: a view of the air force in 2035*. Washington D.C.: Headquarters, United States Air Force.

Hogge, L. J. (2012). *Effective measurement of reliability of repairable USAF systems* (master's thesis), Air Force Institute of Technology. Wright-Patterson AFB, OH, Graduate School of Engineering and Management.

Hurvich, C. M. & Tsai, C.L. (1989). Regression and time series model selection in small samples. *Biometrika*, **76**: 297–307, doi:10.1093/biomet/76.2.297

International Electrotechnical Commission (2006). International standard IEC 61165: application of markov techniques. *IEC Technical Committee 56: Dependability, Commission Electrotechnique Internationale*. Geneva, CH.

Jirutitijaroen, P., & Singh, C. (2004). The effect of transformer maintenance parameters on reliability and cost: a probabilistic model. *Electric Power Systems Research*, 72(3), 213-224.

Kalbfleisch, J.D. & Prentice, R.L. (1980). *The statistical analysis of failure time data*. John Wiley & Sons, Hoboken, NJ.

Kean, J., & Carr, S. (2013). A brief history of early unmanned aircraft. *John's Hopkins APL Technical Digest*, Vol. 32. Number 3. Retrieved on July 20, 2016 from http://www.jhuapl.edu/techdigest/TD/td3203/32_03-Keane.pdf

Keller, J. (2015). Air Force ready to approach industry for enabling technologies in affordable attack drones. *Military Aerospace*. Retrieved from, http://www.militaryaerospace.com/articles/2015/06/inexpensive-attack-drones.html

Kumar, D.U. (1999). New trends in aircraft reliability and maintenance measures. *Journal of Quality in Maintenance Engineering*, 5(4), 287-295.

Kumar, D.U., Knezevic, J., & Crocker, J. (1999). Maintenance free operating period – an alternative measure to MTBF and failure rate for specifying reliability?. *Reliability Engineering and System Safety* 64, 127-131.

Mehta, A. (2015). *Work Outlines Key Steps in Third Offset Tech Development*. Retrieved July 20, 2016, from http://www.defensenews.com/story/defense/innovation/2015/12/14/work-third-offset-tech-development-pentagon-russia/77283732/

Meeker, W.Q & Escobar, L.A. (1998). *Statistical methods for reliability data*. Wiley series in probability and statistics. John Wiley & Sons, New York, NY.

MITRE. (2014). *MITRE: Systems Engineering Guide*. (2014). p. 665, Bedford, MA: MITRE Corporation.

Nicholas, J.M. & Steyn, H (2008). *Project management for business, engineering, and technology: principles and practice (3rd ed.).* Butterworth-Heinemann, Oxford, GB.

O'Connor, P.T. (2002). Practical reliability engineering (4th ed.). John Wiley & Sons, New York.

Pan, J. (1999). *Software reliability*. Carnegie Mellon Univerisity. Dependable Embedded Systems. Retrieved on 11 November 2016, from https://users.ece.cmu.edu/~koopman/des_s99/sw_reliability/

Putter, H., Fiocco, M, & Geskus, R. (2007). Tutorial in biostatistics: competing risks and multi-state models. *Statistics in Medicine*, vol. 26.11, p. 2389-2430.

Reliability Analysis Center. (2003) The applicability of markov analysis methods to reliability, maintainability, and safety. *Selected Topics in Assurance Related Technologies (START)* vol. 10.2. Alion Science. Rome, NY.

Relf, M. N. (1999). Maintenance-free operating periods—the designer's challenge. *Quality and reliability engineering international*, *15*(2), 111-116.

Rigdon, S. E., & Basu, A. P. (2000). *Statistical methods for the reliability of Repairable systems* (1st ed.). United States: Wiley, John & Sons.

Sahner, R. A., & Trivedi, K. S. (1987). Reliability modeling using SHARPE. *IEEE Transactions on Reliability, 36*(2), 186-193.

Secretary of the Air Force. (2003). *AFI 21-118: Improving air and space equipment reliability and maintainability*, Washington DC, Washington DC: Secretary of the Air Force.

Schemmer, B. (1982). Foreword. W. Wagner, *Lightning bugs and other reconnaissance drones*. (p. i-iv). Fallbrook, CA: Armed Forces Journal International.

Spedicato, G.A., Kang, T.S., Yalamanchi, S.B., Mildenberger, T., & Yadav, D. (2016) 'markovchain' package. Easy handling discrete time markov chains. R package version 0.6.5.1.

Systems Engineering Book of Knowledge (n.d.) *Design life (Glossary)*. (n.d.). Retrieved July 20, 2015, from http://sebokwiki.org/wiki/ Design_Life_(glossary)

Tiassou, K., Kanoun, K., Kaâniche, M., Seguin, C., & Papadopoulos, C. (2013). Aircraft operational reliability—A model-based approach and a case study. *Reliability Engineering & System Safety, 120*, 163-176.

Thirtle, M., Johnson, R., & Birkler, J. (1997). The Predator ACTD: A Case Study for Transition Planning to the Formal Acquisition Process. *National Defense Research Institute, RAND*. Retrieved July 20, 2016, from http://www.rand.org/content/dam/rand/pubs/monograph_reports/2005/MR899.pdf

Under Secretary of Defense Acquisition, Technology, and Logistics. (2005). *DoD guide for achieving reliability, availability, and maintainability*, Washington DC: Office of Under Secretary of Defense.

Usher, J. S. (1993). Case Study: Reliability models and misconceptions. *Quality Engineering, 6*(2), 261-271.

Walker, D. (2016). *Fiscal year 2017 air force science and technology*. U.S. House of Representatives Subcommittee on Emerging Threats and Capabilities. Retrieved August 9, 2016, from http://defenseinnovationmarketplace.mil/resources/HHRG-114-AS26-Wstate-WalkerD-20160224.pdf

Walter, N.A. & Watson, P. M. (1971). *Component failure rates and their role in reliability prediction.* Admiralty Surface Weapons Establishment. Portsmouth. Retrieved from DTIC on August 8, 2016 from http://www.dtic.mil/dtic/tr/fulltext/u2/737253.pdf

Yang, G. (2007). Life Cycle Reliability Engineering. John Wiley & Sons, Inc. Hoboken, NJ.

Zhou, H., Li, F., Hu, B., & Huang, W. (2012). A Simulation model for evaluating phased mission system reliability. In *Advances in Technology and Management* (pp. 255-262). Springer Berlin Heidelberg.

Vita

Bryan Bentz
2950 Hobson Way, Wright-Patterson AFB OH 45433
bryan.bentz.1@us.af.mil

EDUCATION

MS in Systems Engineering **2015-present**
United States Air Force Institute of Technology
Wright-Patterson AFB, OH
Thesis title: "Reliability and Cost Impacts for Attritable Systems"
Committee: John Colombi, PhD (Chair), Jason Freels, PhD, Jason Sutherlin

BS in Aeronautical Engineering **2013**
United States Air Force Academy, CO

EXPERIENCE

Systems Engineer/ Test Manager **2013-2015**
Air Force Life Cycle Management Center, Agile Combat Support Directorate, Human Systems Division, Aircrew Performance Branch (AFLCMC/WNUV)
Wright-Patterson AFB, OH

PUBLICATIONS

Forthcoming from IEEE SysCon, 2017

Colombi, J., Bentz, B., Recker, R., Lucas, B., Freels, J. (2017) Attritable Design Trades: Reliability and Cost Implications for Emerging Systems. *2017 SysCon*. Accepted and forthcoming.

Appendix A: Data Analysis in R

Data_Analysis_Thesis_Turn_in.R

bry_b

Sun Jan 29 14:28:37 2017

```
#load necessary libraries
library(SMRD)
library(etm)

library(cmprsk)

library(MASS)

library(markovchain)

library(exptest)

library(fitdistrplus)

library(diagram)

#######################
# Purpose: Data Input
# inputs: N/A
# outputs: CIF of all subsystems, Figure 13
# Author: Bryan Bentz and Joe Berry
# notes: Please read documentation on 'cmprsk' for background
#######################

ftime <- c(rep(1,313),
           rep(2,227),
           rep(3,165),
           rep(4,116),
           rep(5,79),
           rep(6,52),
           rep(7,37),
           rep(8,29),
           rep(9,22),
           rep(10,15))

#taken one past the sortie number where there is less than 20 in population

#using fail codes 1-8
fstatus <- c(#flight 1
  rep(1,9),rep(2,13),rep(3,8),rep(4,1), rep(5,3),rep(6,4),rep(7,0),rep(8,275),
  #flight 2
  rep(1,14),rep(2,7),rep(3,5),rep(4,0),rep(5,6),rep(6,0),rep(7,1),rep(8,194),
  #flight 3
  rep(1,4),rep(2,5),rep(3,3),rep(4,0),rep(5,4),rep(6,0),rep(7,1),rep(8,148),
  #flight 4
  rep(1,5),rep(2,6),rep(3,5),rep(4,1),rep(5,0),rep(6,1),rep(7,1),rep(8,97),
  #flight 5
  rep(1,5),rep(2,3),rep(3,1),rep(4,0),rep(5,0),rep(6,0),rep(7,0),rep(8,70),
  #flight 6
  rep(1,3),rep(2,3),rep(3,1),rep(4,0),rep(5,0),rep(6,1),rep(7,0),rep(8,44),
  #flight 7
  rep(1,2),rep(2,1),rep(3,1),rep(4,0),rep(5,1),rep(6,0),rep(7,0),rep(8,32),
  #flight 8
  rep(1,2),rep(2,0),rep(3,1),rep(4,1),rep(5,1),rep(6,0),rep(7,0),rep(8,24),
  #flight 9
  rep(1,1),rep(2,1),rep(3,1),rep(4,0),rep(5,0),rep(6,0),rep(7,0),rep(8,19),
  #flight 10
  rep(1,1),rep(2,2),rep(3,0),rep(4,0),rep(5,0),rep(6,0),rep(7,0),rep(8,12)
)

#taken one past the sortie number where there is less than 20 in population
#######################
# Purpose: Calculate and plot cumulativie incidence function
# inputs: data above and 'cmprsk'library of R
# outputs: CIF plot,CIF$`1 i`$est[n],CIF$`1 i`$var[n]
# Author: Bryan Bentz and Joe Berry
#######################

CIF <- cuminc(ftime=ftime,fstatus=fstatus,cencode = 8)

print.cuminc(CIF, ntp=22)

plot(CIF, curvlab = c("Electronics", "Fuel Management", "Launcher",
"Operator",
                      "Propulsion", "Recovery","Structure"),main="Cumulative
Incidence Functions of
     Failure Modes", xlab="Sorties",
     ylab="Probability",ylim = c(0, 0.16),
     ci.type = "pointwise", col = c(1:6,8), lwd = par('lwd'), lty =
1:length(CIF), cex = 0.6)

xticks <- seq(0, 10, 1)
yticks <- seq(0, 0.16, 0.05)
axis(1, at = xticks, labels = xticks, las=1, tck=-.01)
```

```
#######################
# Purpose: calculate the hazards rates of subsystem CIF and Conf Intervals
# inputs: CIF estimates and variances
# outputs: haz1-haz7, plots of hazard rates with CIs (Figures 14-20 in Thesis
Doc)
# notes: CIs based on logit transformation of normal approximation
# Author: Bryan Bentz and Joe Berry
#######################

HazConfInt <- function(est,var,times) {
  i=1
  haz=rep(0,10)
  hazup=rep(0,10)
  hazlow=rep(0,10)
  while (i<length(times)) {

    haz[times[i+1]]=(est[i+2]-est[i])/(1-est[i])

    w2 = exp(1.96*sqrt(var[i+2])/(est[i+2]*(1-est[i+2])))
    w1 = exp(1.96*sqrt(var[i])/(est[i]*(1-est[i])))

    F2up = est[i+2]/(est[i+2]+(1-est[i+2])/w2)
    F1up = est[i]/(est[i]+(1-est[i])/w2)

    hazup[times[i+1]] = (F2up - F1up) / (1-F1up)

    F2low = est[i+2]/(est[i+2]+(1-est[i+2])*w2)
    F1low = est[i]/(est[i]+(1-est[i])*w2)

    hazlow[times[i+1]] = (F2low - F1low) / (1-F1low)

    i=i+2
  }#while

  hazconfint <- data.frame(haz,hazlow,hazup)

  return(hazconfint)
}

haz1 <- HazConfInt(CIF$`1 1`$est,CIF$`1 1`$var,CIF$`1 1`$time)
par(col='black')
yupper=.05
plot(haz1$haz,ylim=c(0,yupper),main="Failure Mode 1 Hazard from CIF",
     xlab='Sortie',ylab='Hazard Rate',xlim=c(1,10),axes=F, pch = 19)
xticks <- seq(0, 10, 1)
yticks <- seq(0,0.05,0.01)
axis(1, at = xticks, labels = xticks, las=1, tck=0)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(haz1$hazlow,ylim=c(0,yupper),ann=F,xlim=c(1,10),axes=F, pch = 24)
par(new=TRUE)
plot(haz1$hazup,ylim=c(0,yupper),ann=F,xlim=c(1,10),axes=F, pch = 25)
```

```
par(new=FALSE)

haz2 <- HazConfInt(CIF$`1 2`$est,CIF$`1 2`$var,CIF$`1 2`$time)
par(col='black')
plot(haz2$haz,ylim=c(0,yupper),main="Failure Mode 2 Hazard from CIF",
     xlab='Sortie',ylab='Hazard Rate',xlim=c(1,10),axes=F, pch = 19)
xticks <- seq(0, 10, 1)
yticks <- seq(0,0.05,0.01)
axis(1, at = xticks, labels = xticks, las=1, tck=-.01)
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(haz2$hazlow,ylim=c(0,yupper),ann=F,xlim=c(1,10),axes=F, pch = 24)
```

```
par(new=TRUE)
plot(haz2$hazup,ylim=c(0,yupper),ann=F,xlim=c(1,10),axes=F, pch = 25)
```

```
par(new=FALSE)

haz3 <- HazConfInt(CIF$`1 3`$est,CIF$`1 3`$var,CIF$`1 3`$time)
yupper=0.05 # was 0.3
par(col='black')
plot(haz3$haz,ylim=c(0,yupper),main="Failure Mode 3 Hazard from CIF",
     xlab='Sortie',ylab='Hazard Rate',xlim=c(1,10),axes=F, pch = 19)
xticks <- seq(0, 10, 1)
yticks <- seq(0,0.05,0.01)
axis(1, at = xticks, labels = xticks, las=1, tck=-.01)
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(haz3$hazlow,ylim=c(0,yupper),ann=F,xlim=c(1,10),axes=F, pch = 24)
par(new=TRUE)
plot(haz3$hazup,ylim=c(0,yupper),ann=F,xlim=c(1,10),axes=F, pch = 25)
```

```
par(new=FALSE)

haz4 <- HazConfInt(CIF$`1 4`$est,CIF$`1 4`$var,CIF$`1 4`$time)
yupper=0.05
par(col='black')
plot(haz4$haz,ylim=c(0,yupper),main="Failure Mode 4 Hazard from CIF",
     xlab='Sortie',ylab='Hazard Rate',xlim=c(1,10),axes=F, pch = 19)
xticks <- seq(0, 10, 1)
yticks <- seq(0,0.05,0.01)
axis(1, at = xticks, labels = xticks, las=1, tck=-.01)
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(haz4$hazlow,ylim=c(0,yupper),ann=F,xlim=c(1,10),axes=F, pch = 24)
par(new=TRUE)
plot(haz4$hazup,ylim=c(0,yupper),ann=F,xlim=c(1,10),axes=F, pch = 25)
```

```
par(new=FALSE)

haz5 <- HazConfInt(CIF$`1 5`$est,CIF$`1 5`$var,CIF$`1 5`$time)
par(col='black')
plot(haz5$haz,ylim=c(0,yupper),main="Failure Mode 5 Hazard from CIF",
     xlab='Sortie',ylab='Hazard Rate',xlim=c(1,10),axes=F, pch = 19)
xticks <- seq(0, 10, 1)
yticks <- seq(0,0.05,0.01)
axis(1, at = xticks, labels = xticks, las=1, tck=-.01)
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(haz5$hazlow,ylim=c(0,yupper),ann=F,xlim=c(1,10),axes=F, pch = 24)
```

```
par(new=TRUE)
plot(haz5$hazup,ylim=c(0,yupper),ann=F,xlim=c(1,10),axes=F, pch = 25)
```

```
par(new=FALSE)

haz6 <- HazConfInt(CIF$`1 6`$est,CIF$`1 6`$var,CIF$`1 6`$time)
par(col='black')
plot(haz6$haz,ylim=c(0,yupper),main="Failure Mode 6 Hazard from CIF",
     xlab='Sortie',ylab='Hazard Rate',xlim=c(1,10),axes=F, pch = 19)
xticks <- seq(0, 10, 1)
yticks <- seq(0,0.05,0.01)
axis(1, at = xticks, labels = xticks, las=1, tck=-.01)
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(haz6$hazlow,ylim=c(0,yupper),ann=F,xlim=c(1,10),axes=F, pch = 24)
par(new=TRUE)
plot(haz6$hazup,ylim=c(0,yupper),ann=F,xlim=c(1,10),axes=F, pch = 25)
```

```
par(new=FALSE)

haz7 <- HazConfInt(CIF$`1 7`$est,CIF$`1 7`$var,CIF$`1 7`$time)
par(col='black')
plot(haz7$haz,ylim=c(0,yupper),main="Failure Mode 7 Hazard from CIF",
     xlab='Sortie',ylab='Hazard Rate',xlim=c(1,10),axes=F, pch = 19)
xticks <- seq(0, 10, 1)
yticks <- seq(0,0.05,0.01)
axis(1, at = xticks, labels = xticks, las=1, tck=-.01)
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(haz7$hazlow,ylim=c(0,yupper),ann=F,xlim=c(1,10),axes=F, pch = 24)
par(new=TRUE)
plot(haz7$hazup,ylim=c(0,yupper),ann=F,xlim=c(1,10),axes=F, pch = 25)

par(new=FALSE)
#######################
# Purpose: analyze CIF$`1 1-7`$est to determine which distribution fits CIFs
# inputs: CIF estimates and variances
# outputs: Q-Q plots to determine whether CIFs fit exponential distribution
# notes: CIs based on normal approximation
# Author: Bryan Bentz
#######################
fm1 <- c(CIF$`1 1`$est[3],CIF$`1 1`$est[5],CIF$`1 1`$est[7],CIF$`1 1`$est[9],
         CIF$`1 1`$est[11],CIF$`1 1`$est[13],CIF$`1 1`$est[15],CIF$`1
1`$est[17],CIF$`1 1`$est[19],
         CIF$`1 1`$est[21]) #requires callouts as cmprsk draws step function
with end and start points
qqplot(x=qexp(ppoints(length(fm1))), y=fm1, main="Exponential Q-Q Plot",
       xlab="Theoretical Exponential Quantiles", ylab="Sample Quantiles")
qqline(fm1, distribution=qexp, col="blue",lty=2)
```

```
#######################

fm2 <- c(CIF$`1 2`$est[3],CIF$`1 2`$est[5],CIF$`1 2`$est[7],CIF$`1 2`$est[9],
         CIF$`1 2`$est[11],CIF$`1 2`$est[13],CIF$`1 2`$est[15],CIF$`1
2`$est[17],CIF$`1 2`$est[19])
qqplot(x=qexp(ppoints(length(fm2))), y=fm2, main="Exponential Q-Q Plot",
       xlab="Theoretical Exponential Quantiles", ylab="Sample Quantiles")
qqline(fm2, distribution=qexp, col="blue",lty=2)

#######################

fm3 <- c(CIF$`1 3`$est[3],CIF$`1 3`$est[5],CIF$`1 3`$est[7],CIF$`1 3`$est[9],
         CIF$`1 3`$est[11],CIF$`1 3`$est[13],CIF$`1 3`$est[15],CIF$`1
3`$est[17])
qqplot(x=qexp(ppoints(length(fm3))), y=fm3, main="Exponential Q-Q Plot",
       xlab="Theoretical Quantiles", ylab="Sample Quantiles")
qqline(fm3, distribution=qexp, col="blue",lty=2)

#######################

fm4 <- c(CIF$`1 4`$est[3],CIF$`1 4`$est[5],CIF$`1 4`$est[7])
qqplot(x=qexp(ppoints(length(fm4))), y=fm4, main="Exponential Q-Q Plot",
       xlab="Theoretical Quantiles", ylab="Sample Quantiles")
qqline(fm4, distribution=qexp, col="blue",lty=2)

#######################

fm5 <- c(CIF$`1 5`$est[3],CIF$`1 5`$est[5],CIF$`1 5`$est[7],CIF$`1 5`$est[9],
         CIF$`1 5`$est[11])
qqplot(x=qexp(ppoints(length(fm5))), y=fm5, main="Exponential Q-Q Plot",
       xlab="Theoretical Quantiles", ylab="Sample Quantiles")
qqline(fm5, distribution=qexp, col="blue",lty=2)

#######################

fm6 <- c(CIF$`1 6`$est[3],CIF$`1 6`$est[5],CIF$`1 6`$est[7])
qqplot(x=qexp(ppoints(length(fm6))), y=fm6, main="Exponential Q-Q Plot",
       xlab="Theoretical Quantiles", ylab="Sample Quantiles")
qqline(fm6, distribution=qexp, col="blue",lty=2)

#######################

fm7 <- c(CIF$`1 7`$est[3],CIF$`1 7`$est[5],CIF$`1 7`$est[7])
qqplot(x=qexp(ppoints(length(fm7))), y=fm7, main="Exponential Q-Q Plot",
       xlab="Theoretical Quantiles", ylab="Sample Quantiles")
qqline(fm7, distribution=qexp, col="blue",lty=2)

#######################
# Purpose: Determine if a distribution that varies with time would better fit
# inputs: arrays of fm(i) for all 7 subsystems
# outputs: Determination of each failure mode's failure-time data relationship
to Expon. Dist.
# notes: to avoid overfitting, the AIC penalizes for the amount of
distribution parameters.
```

```
#        the AICc corrects AIC for the small sample size.
#        fit(i)exp$estimate and fit(i)weib$estimate are not used in the
future.  Only to determine
#        if hazard rate vary with time.  See section 4.2 of Thesis doc.
# Author: Bryan Bentz
#######################

exponparam <-1 #free parameters of exponential distribution
weibparam <-2 #free parameters of weibull distribution

#######################

fit1exp <- fitdistr(fm1, "exponential")
fit1weib <- fitdistr(fm1, "weibull")

## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced

## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced

## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced

## Warning in densfun(x, parm[1], parm[2], ...): NaNs produced

AICexp1 <- 2*exponparam-2*fit1exp$loglik
AICexp1c <- AICexp1 + (2*(exponparam+1)*(exponparam+2))/(length(fm1)-
exponparam-2)
AICweib1 <- 2*weibparam-2*fit1weib$loglik
AICweib1c <- AICweib1 + (2*(weibparam+1)*(weibparam+2))/length(fm1)-weibparam-
2
AICdiff1 <- ((AICweib1c-AICexp1c)/AICweib1c)*100

#######################

fit2exp <- fitdistr(fm2, "exponential")
fit2weib <- fitdistr(fm2, "weibull")

AICexp2 <- 2*exponparam-2*fit2exp$loglik
AICexp2c <- AICexp2 + (2*(exponparam+1)*(exponparam+2))/(length(fm2)-
exponparam-2)
AICweib2 <- 2*weibparam-2*fit2weib$loglik
AICweib2c <- AICweib2 + (2*(weibparam+1)*(weibparam+2))/length(fm2)-weibparam-
2
AICdiff2 <- ((AICweib2c-AICexp2c)/AICweib2c)*100

#######################

fit3exp <- fitdistr(fm3, "exponential")
fit3weib <- fitdistr(fm3, "weibull")

AICexp3 <- 2*exponparam-2*fit3exp$loglik
AICexp3c <- AICexp3 + (2*(exponparam+1)*(exponparam+2))/(length(fm3)-
exponparam-2)
AICweib3 <- 2*weibparam-2*fit3weib$loglik
AICweib3c <- AICweib3 + (2*(weibparam+1)*(weibparam+2))/length(fm3)-weibparam-
```

```
2
AICdiff3 <- ((AICweib3c-AICexp3c)/AICweib3c)*100

#######################

fit4exp <- fitdistr(fm4, "exponential")
fit4weib <- fitdistr(fm4, "weibull")

AICexp4 <- 2*exponparam-2*fit4exp$loglik
AICexp4c <- AICexp4 + (2*(exponparam+1)*(exponparam+2))/(length(fm4)-
exponparam-1)
AICweib4 <- 2*weibparam-2*fit4weib$loglik
AICweib4c <- AICweib4 + (2*(weibparam+1)*(weibparam+2))/length(fm4)-weibparam-
2
AICdiff4 <- ((AICweib4c-AICexp4c)/AICweib4c)*100

#######################

fit5weib <- fitdistr(fm5, "weibull")

AICweib5 <- 2*weibparam-2*fit5weib$loglik
AICweib5c <- AICweib5 + (2*(weibparam+1)*(weibparam+2))/length(fm5)-weibparam-
2

fit5exp <- fitdistr(fm5, "exponential")
AICexp5 <- 2*exponparam-2*fit5exp$loglik
AICexp5c <- AICexp5 + (2*(exponparam+1)*(exponparam+2))/(length(fm5)-
exponparam-2)

AICdiff5 <- ((AICweib5c-AICexp5c)/AICweib5c)*100

#######################

fit6exp <- fitdistr(fm6, "exponential")
fit6weib <- fitdistr(fm6, "weibull")

AICexp6 <- 2*exponparam-2*fit6exp$loglik
AICexp6c <- AICexp6 + (2*(exponparam+1)*(exponparam+2))/(length(fm6)-
exponparam-1)
AICweib6 <- 2*weibparam-2*fit6weib$loglik
AICweib6c <- AICweib6 + (2*(weibparam+1)*(weibparam+2))/length(fm6)-weibparam-
2
AICdiff6 <- ((AICweib6c-AICexp6c)/AICweib6c)*100

#######################

fm7 <- c(CIF$`1 7`$est[3],CIF$`1 7`$est[5],CIF$`1 7`$est[7])

fit7weib <- fitdistr(fm7, "weibull")

AICweib7 <- 2*weibparam-2*fit7weib$loglik
AICweib7c <- AICweib7 + (2*(weibparam+1)*(weibparam+2))/(length(fm7)-
weibparam-2)
```

```
fit7exp <- fitdistr(fm7, "exponential")
AICexp7 <- 2*exponparam-2*fit7exp$loglik
AICexp7c <- AICexp7 + (2*(exponparam+1)*(exponparam+2))/(length(fm7)-
exponparam-3)

AICdiff7 <- ((AICweib7c-AICexp7c)/AICweib7c)*100

lamdafm7est <- 1/fit7exp$estimate #fm7 hazard rate. Table 5 of thesis puts
AICc "exp" as best fit

#logit transformation based on 3.16, p. 57 in Meeker & Escobar.  See thesis
for citation
fm7low <- c((CIF$`1 7`$est[3]/(CIF$`1 7`$est[3]+(1-CIF$`1 7`$est[3])*
                     exp((1.96*sqrt(CIF$`1 7`$var[3])/(CIF$`1 7`$est[3]*(1-
CIF$`1 7`$est[3])))))),
          (CIF$`1 7`$est[5]/(CIF$`1 7`$est[5]+(1-CIF$`1 7`$est[5])*
                     exp((1.96*sqrt(CIF$`1 7`$var[5])/(CIF$`1 7`$est[5]*(1-
CIF$`1 7`$est[5])))))),
          (CIF$`1 7`$est[7]/(CIF$`1 7`$est[7]+(1-CIF$`1 7`$est[7])*
                               exp((1.96*sqrt(CIF$`1 7`$var[7])/(CIF$`1
7`$est[7]*(1-CIF$`1 7`$est[7])))))))

fit7lowexp <- fitdistr(fm7low, "exponential")
lamdafm7low <- 1/fit7lowexp$estimate

#logit transformation based on 3.16, p. 57 in Meeker & Escobar.  See thesis
for citation
fm7up <- c((CIF$`1 7`$est[3]/(CIF$`1 7`$est[3]+(1-CIF$`1 7`$est[3])/
                                 exp((1.96*sqrt(CIF$`1 7`$var[3])/(CIF$`1
7`$est[3]*(1-CIF$`1 7`$est[3])))))),
            (CIF$`1 7`$est[5]/(CIF$`1 7`$est[5]+(1-CIF$`1 7`$est[5])/
                                 exp((1.96*sqrt(CIF$`1 7`$var[5])/(CIF$`1
7`$est[5]*(1-CIF$`1 7`$est[5])))))),
            (CIF$`1 7`$est[7]/(CIF$`1 7`$est[7]+(1-CIF$`1 7`$est[7])/
                                 exp((1.96*sqrt(CIF$`1 7`$var[7])/(CIF$`1
7`$est[7]*(1-CIF$`1 7`$est[7])))))))

fit7upexp <- fitdistr(fm7up, "exponential")
lamdafm7up <- 1/fit7upexp$estimate

#######################
# Purpose: create information for baseline Markov Chain Models (MCM)
# inputs: haz1-haz7
# outputs: recursion(1-9), mcLCASD, finalStatet1-9
# Author: Bryan Bentz
#######################

step <- 1 #this will be used calculate the final state probabilities
initialStatet1 <- c(1,0,0,0,0,0,0,0,0) #assumes that every unit is operational
upon delivery, t0

#Creates the probability of "regeneration" for the nominal baseline Design
recursion1 <- 1-
```

```
(haz1$haz[1]+haz2$haz[1]+haz3$haz[1]+haz4$haz[1]+haz5$haz[1]+haz6$haz[1]+lamda
fm7est)
recursion2 <- 1-
(haz1$haz[2]+haz2$haz[2]+haz3$haz[2]+haz4$haz[2]+haz5$haz[2]+haz6$haz[2]+lamda
fm7est)
recursion3 <- 1-
(haz1$haz[3]+haz2$haz[3]+haz3$haz[3]+haz4$haz[3]+haz5$haz[3]+haz6$haz[3]+lamda
fm7est)
recursion4 <- 1-
(haz1$haz[4]+haz2$haz[4]+haz3$haz[4]+haz4$haz[4]+haz5$haz[4]+haz6$haz[4]+lamda
fm7est)
recursion5 <- 1-
(haz1$haz[5]+haz2$haz[5]+haz3$haz[5]+haz4$haz[5]+haz5$haz[5]+haz6$haz[5]+lamda
fm7est)
recursion6 <- 1-
(haz1$haz[6]+haz2$haz[6]+haz3$haz[6]+haz4$haz[6]+haz5$haz[6]+haz6$haz[6]+lamda
fm7est)
recursion7 <- 1-
(haz1$haz[7]+haz2$haz[7]+haz3$haz[7]+haz4$haz[7]+haz5$haz[7]+haz6$haz[7]+lamda
fm7est)
recursion8 <- 1-
(haz1$haz[8]+haz2$haz[8]+haz3$haz[8]+haz4$haz[8]+haz5$haz[8]+haz6$haz[8]+lamda
fm7est)
recursion9 <- 1-
(haz1$haz[9]+haz2$haz[9]+haz3$haz[9]+haz4$haz[9]+haz5$haz[9]+haz6$haz[9]+lamda
fm7est)

#######################
# Purpose: create discrete time non-homogeneous Markov Chain Models (MCM) of
baseline design at est.
# inputs: recursion [i], haz1-haz7, probabilities of transition from "failed"
to destroyed
# outputs: mcLCASD, a discrete time non-homogeneous MCM
# Author: Bryan Bentz
#######################

stateNames <- c("Operational", "Electronics Bay", "Fuel
Management","Launcher","Operator",
                "Propulsion","Recovery","Structure","Destroyed")
P1 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion1,haz1$haz[1],haz2$haz[1],haz3$haz[1],haz4$haz[1],haz5$haz[1
],haz6$haz[1],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (12/13),0,0,0,0,0,0,0,(1/13),
                     .75,0,0,0,0,0,0,0,.25,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     .75,0,0,0,0,0,0,0,.25,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t1")
P2 <- new("markovchain", states = stateNames, transitionMatrix =
matrix(c(recursion2,haz1$haz[2],haz2$haz[2],haz3$haz[2],haz4$haz[2],haz5$haz[2
],haz6$haz[2],lamdafm7est, 0,
                     (13/14),0,0,0,0,0,0,0,(1/14),
                     (6/7),0,0,0,0,0,0,0,(1/7),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (5/6),0,0,0,0,0,0,0,(1/6),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t2")

P3 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion3,haz1$haz[3],haz2$haz[3],haz3$haz[3],haz4$haz[3],haz5$haz[3
],haz6$haz[3],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t3")

P4 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion4,haz1$haz[4],haz2$haz[4],haz3$haz[4],haz4$haz[4],haz5$haz[4
],haz6$haz[4],lamdafm7est, 0,
                     (4/5),0,0,0,0,0,0,0,(1/5),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t4")

P5 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion5,haz1$haz[5],haz2$haz[5],haz3$haz[5],haz4$haz[5],haz5$haz[5
],haz6$haz[5],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t5")

P6 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion6,haz1$haz[6],haz2$haz[6],haz3$haz[6],haz4$haz[6],haz5$haz[6
],haz6$haz[6],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t6")

P7 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion7,haz1$haz[7],haz2$haz[7],haz3$haz[7],haz4$haz[7],haz5$haz[7
],haz6$haz[7],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t7")

P8 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion8,haz1$haz[8],haz2$haz[8],haz3$haz[8],haz4$haz[8],haz5$haz[8
],haz6$haz[8],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t8")

P9 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion9,haz1$haz[9],haz2$haz[9],haz3$haz[9],haz4$haz[9],haz5$haz[9
],haz6$haz[9],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t9")

mcLCASD <- new("markovchainList",markovchains =
list(P1,P2,P3,P4,P5,P6,P7,P8,P9),
               name = "Attritable System Behavior")

#calculating the probability of entering each state for baseline design
finalStatet1 <- initialStatet1*mcLCASD[[1]]^step
finalStatet2 <- finalStatet1*mcLCASD[[2]]^step
finalStatet3 <- finalStatet2*mcLCASD[[3]]^step
finalStatet4 <- finalStatet3*mcLCASD[[4]]^step
finalStatet5 <- finalStatet4*mcLCASD[[5]]^step
finalStatet6 <- finalStatet5*mcLCASD[[6]]^step
finalStatet7 <- finalStatet6*mcLCASD[[7]]^step
finalStatet8 <- finalStatet7*mcLCASD[[8]]^step
finalStatet9 <- finalStatet8*mcLCASD[[9]]^step

#######################
# Purpose: create information for lower estimate Markov Chain Model (MCM)
# inputs: haz1-haz7
# outputs: recursion(1-9)low,mcLCASDlow, FinalStatelowt1-9 because there's one
for each time step.
# Author: Bryan Bentz
#######################


#Creates the probability of "regeneration" for the lower 95% estimate of the
baseline Design
recursion1low <- 1-
(haz1$hazlow[1]+haz2$hazlow[1]+haz3$hazlow[1]+haz4$hazlow[1]+haz5$hazlow[1]+ha
z6$hazlow[1]+lamdafm7low)
recursion2low <- 1-
(haz1$hazlow[2]+haz2$hazlow[2]+haz3$hazlow[2]+haz4$hazlow[2]+haz5$hazlow[2]+ha
z6$hazlow[2]+lamdafm7low)
recursion3low <- 1-
(haz1$hazlow[3]+haz2$hazlow[3]+haz3$hazlow[3]+haz4$hazlow[3]+haz5$hazlow[3]+ha
z6$hazlow[3]+lamdafm7low)
recursion4low <- 1-
(haz1$hazlow[4]+haz2$hazlow[4]+haz3$hazlow[4]+haz4$hazlow[4]+haz5$hazlow[4]+ha
z6$hazlow[4]+lamdafm7low)
recursion5low <- 1-
(haz1$hazlow[5]+haz2$hazlow[5]+haz3$hazlow[5]+haz4$hazlow[5]+haz5$hazlow[5]+ha
z6$hazlow[5]+lamdafm7low)
recursion6low <- 1-
(haz1$hazlow[6]+haz2$hazlow[6]+haz3$hazlow[6]+haz4$hazlow[6]+haz5$hazlow[6]+ha
z6$hazlow[6]+lamdafm7low)
recursion7low <- 1-
(haz1$hazlow[7]+haz2$hazlow[7]+haz3$hazlow[7]+haz4$hazlow[7]+haz5$hazlow[7]+ha
z6$hazlow[7]+lamdafm7low)
recursion8low <- 1-
```

```
(haz1$hazlow[8]+haz2$hazlow[8]+haz3$hazlow[8]+haz4$hazlow[8]+haz5$hazlow[8]+ha
z6$hazlow[8]+lamdafm7low)
recursion9low <- 1-
(haz1$hazlow[9]+haz2$hazlow[9]+haz3$hazlow[9]+haz4$hazlow[9]+haz5$hazlow[9]+ha
z6$hazlow[9]+lamdafm7low)

stateNames <- c("Operational", "Electronics Bay", "Fuel
Management","Launcher","Operator",
                "Propulsion","Recovery","Structure","Destroyed")
Q1 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion1low,haz1$hazlow[1],haz2$hazlow[1],haz3$hazlow[1],haz4$hazlo
w[1],haz5$hazlow[1],haz6$hazlow[1],lamdafm7low, 0,
                     1,0,0,0,0,0,0,0,0,
                     (12/13),0,0,0,0,0,0,0,(1/13),
                     .75,0,0,0,0,0,0,0,.25,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     .75,0,0,0,0,0,0,0,.25,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t1")
Q2 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion2low,haz1$hazlow[2],haz2$hazlow[2],haz3$hazlow[2],haz4$hazlo
w[2],haz5$hazlow[2],haz6$hazlow[2],lamdafm7low, 0,
                     (13/14),0,0,0,0,0,0,0,(1/14),
                     (6/7),0,0,0,0,0,0,0,(1/7),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (5/6),0,0,0,0,0,0,0,(1/6),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t2")

Q3 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion3low,haz1$hazlow[3],haz2$hazlow[3],haz3$hazlow[3],haz4$hazlo
w[3],haz5$hazlow[3],haz6$hazlow[3],lamdafm7low, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t3")

Q4 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion4low,haz1$hazlow[4],haz2$hazlow[4],haz3$hazlow[4],haz4$hazlo
w[4],haz5$hazlow[4],haz6$hazlow[4],lamdafm7low, 0,
                     (4/5),0,0,0,0,0,0,0,(1/5),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t4")

Q5 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion5low,haz1$hazlow[5],haz2$hazlow[5],haz3$hazlow[5],haz4$hazlo
w[5],haz5$hazlow[5],haz6$hazlow[5],lamdafm7low, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t5")

Q6 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion6low,haz1$hazlow[6],haz2$hazlow[6],haz3$hazlow[6],haz4$hazlo
w[6],haz5$hazlow[6],haz6$hazlow[6],lamdafm7low, 0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t6")

Q7 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion7low,haz1$hazlow[7],haz2$hazlow[7],haz3$hazlow[7],haz4$hazlo
w[7],haz5$hazlow[7],haz6$hazlow[7],lamdafm7low, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t7")
```

```
Q8 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion8low,haz1$hazlow[8],haz2$hazlow[8],haz3$hazlow[8],haz4$hazlo
w[8],haz5$hazlow[8],haz6$hazlow[8],lamdafm7low, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t8")

Q9 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion9low,haz1$hazlow[9],haz2$hazlow[9],haz3$hazlow[9],haz4$hazlo
w[9],haz5$hazlow[9],haz6$hazlow[9],lamdafm7low, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t9")

mcLCASDlow <- new("markovchainList",markovchains =
list(Q1,Q2,Q3,Q4,Q5,Q6,Q7,Q8,Q9),
                  name = "Attritable System Behavior at Low Estimate")

#calculating the probability of entering each state for baseline design using
low est of 95%
finalStatet1low <- initialStatet1*mcLCASDlow[[1]]^step
finalStatet2low <- finalStatet1low*mcLCASDlow[[2]]^step
finalStatet3low <- finalStatet2low*mcLCASDlow[[3]]^step
finalStatet4low <- finalStatet3low*mcLCASDlow[[4]]^step
finalStatet5low <- finalStatet4low*mcLCASDlow[[5]]^step
finalStatet6low <- finalStatet5low*mcLCASDlow[[6]]^step
finalStatet7low <- finalStatet6low*mcLCASDlow[[7]]^step
finalStatet8low <- finalStatet7low*mcLCASDlow[[8]]^step
finalStatet9low <- finalStatet8low*mcLCASDlow[[9]]^step

########################
# Purpose: create information for lower estimate Markov Chain Model (MCM)
# inputs: haz1-haz7
# outputs: recursion(1-9)up, mcLCASDup, FinalStateupt1-9 because there's one
for each time step.
# Author: Bryan Bentz
########################

#Creates the probability of "regeneration" for the upper 95% estimate of the
```

```
baseline Design
recursion1up <- 1-
(haz1$hazup[1]+haz2$hazup[1]+haz3$hazup[1]+haz4$hazup[1]+haz5$hazup[1]+haz6$ha
zup[1]+lamdafm7up)
recursion2up <- 1-
(haz1$hazup[2]+haz2$hazup[2]+haz3$hazup[2]+haz4$hazup[2]+haz5$hazup[2]+haz6$ha
zup[2]+lamdafm7up)
recursion3up <- 1-
(haz1$hazup[3]+haz2$hazup[3]+haz3$hazup[3]+haz4$hazup[3]+haz5$hazup[3]+haz6$ha
zup[3]+lamdafm7up)
recursion4up <- 1-
(haz1$hazup[4]+haz2$hazup[4]+haz3$hazup[4]+haz4$hazup[4]+haz5$hazup[4]+haz6$ha
zup[4]+lamdafm7up)
recursion5up <- 1-
(haz1$hazup[5]+haz2$hazup[5]+haz3$hazup[5]+haz4$hazup[5]+haz5$hazup[5]+haz6$ha
zup[5]+lamdafm7up)
recursion6up <- 1-
(haz1$hazup[6]+haz2$hazup[6]+haz3$hazup[6]+haz4$hazup[6]+haz5$hazup[6]+haz6$ha
zup[6]+lamdafm7up)
recursion7up <- 1-
(haz1$hazup[7]+haz2$hazup[7]+haz3$hazup[7]+haz4$hazup[7]+haz5$hazup[7]+haz6$ha
zup[7]+lamdafm7up)
recursion8up <- 1-
(haz1$hazup[8]+haz2$hazup[8]+haz3$hazup[8]+haz4$hazup[8]+haz5$hazup[8]+haz6$ha
zup[8]+lamdafm7up)
recursion9up <- 1-
(haz1$hazup[9]+haz2$hazup[9]+haz3$hazup[9]+haz4$hazup[9]+haz5$hazup[9]+haz6$ha
zup[9]+lamdafm7up)

stateNames <- c("Operational", "Electronics Bay", "Fuel
Management","Launcher","Operator",
                "Propulsion","Recovery","Structure","Destroyed")
R1 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion1up,haz1$hazup[1],haz2$hazup[1],haz3$hazup[1],haz4$hazup[1],
haz5$hazup[1],haz6$hazup[1],lamdafm7up, 0,
                     1,0,0,0,0,0,0,0,0,
                     (12/13),0,0,0,0,0,0,0,(1/13),
                     .75,0,0,0,0,0,0,0,.25,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     .75,0,0,0,0,0,0,0,.25,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t1")
R2 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion2up,haz1$hazup[2],haz2$hazup[2],haz3$hazup[2],haz4$hazup[2],
haz5$hazup[2],haz6$hazup[2],lamdafm7up, 0,
                     (13/14),0,0,0,0,0,0,0,(1/14),
                     (6/7),0,0,0,0,0,0,0,(1/7),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (5/6),0,0,0,0,0,0,0,(1/6),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t2")

R3 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion3up,haz1$hazup[3],haz2$hazup[3],haz3$hazup[3],haz4$hazup[3],
haz5$hazup[3],haz6$hazup[3],lamdafm7up, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t3")

R4 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion4up,haz1$hazup[4],haz2$hazup[4],haz3$hazup[4],haz4$hazup[4],
haz5$hazup[4],haz6$hazup[4],lamdafm7up, 0,
                     (4/5),0,0,0,0,0,0,0,(1/5),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t4")

R5 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion5up,haz1$hazup[5],haz2$hazup[5],haz3$hazup[5],haz4$hazup[5],
haz5$hazup[5],haz6$hazup[5],lamdafm7up, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t5")

R6 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion6up,haz1$hazup[6],haz2$hazup[6],haz3$hazup[6],haz4$hazup[6],
haz5$hazup[6],haz6$hazup[6],lamdafm7up, 0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t6")

R7 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion7up,haz1$hazup[7],haz2$hazup[7],haz3$hazup[7],haz4$hazup[7],
haz5$hazup[7],haz6$hazup[7],lamdafm7up, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t7")

R8 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion8up,haz1$hazup[8],haz2$hazup[8],haz3$hazup[8],haz4$hazup[8],
haz5$hazup[8],haz6$hazup[8],lamdafm7up, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t8")

R9 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion9up,haz1$hazup[9],haz2$hazup[9],haz3$hazup[9],haz4$hazup[9],
haz5$hazup[9],haz6$hazup[9],lamdafm7up, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t9")

mcLCASDup <- new("markovchainList",markovchains =
list(R1,R2,R3,R4,R5,R6,R7,R8,R9),
                 name = "Attritable System Behavior at High Estimate")

#calculating the probability of entering each state for baseline design using
upper est of 95%
finalStatet1up <- initialStatet1*mcLCASDup[[1]]^step
finalStatet2up <- finalStatet1up*mcLCASDup[[2]]^step
finalStatet3up <- finalStatet2up*mcLCASDup[[3]]^step
finalStatet4up <- finalStatet3up*mcLCASDup[[4]]^step
finalStatet5up <- finalStatet4up*mcLCASDup[[5]]^step
finalStatet6up <- finalStatet5up*mcLCASDup[[6]]^step
finalStatet7up <- finalStatet6up*mcLCASDup[[7]]^step
finalStatet8up <- finalStatet7up*mcLCASDup[[8]]^step
finalStatet9up <- finalStatet8up*mcLCASDup[[9]]^step

#######################
# Purpose: create information for altered FM1 Markov Chain Model (MCM)
# inputs: haz1-haz7
# outputs: recursion(1-9)up, mcLCASDnew1, FinalStatenew1t1-9 because there's
one for each time step.
# Author: Bryan Bentz
#######################

Delta1 = 1.0 #ratio of new hazard rate over old hazard rate
stateNames <- c("Operational", "Electronics Bay", "Fuel
Management","Launcher","Operator",
                "Propulsion","Recovery","Structure","Destroyed")
S1 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(Delta1*haz1$haz[1]+haz2$haz[1]+haz3$haz[1]+haz4$haz[1]+haz5$haz[1]+haz6$haz[1
]+lamdafm7est),Delta1*haz1$haz[1],haz2$haz[1],haz3$haz[1],haz4$haz[1],haz5$haz
[1],haz6$haz[1],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (12/13),0,0,0,0,0,0,0,(1/13),
                     .75,0,0,0,0,0,0,0,.25,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     .75,0,0,0,0,0,0,0,.25,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t1")
S2 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(Delta1*haz1$haz[2]+haz2$haz[2]+haz3$haz[2]+haz4$haz[2]+haz5$haz[2]+haz6$haz[2
]+lamdafm7est),Delta1*haz1$haz[2],haz2$haz[2],haz3$haz[2],haz4$haz[2],haz5$haz
[2],haz6$haz[2],lamdafm7est, 0,
                     (13/14),0,0,0,0,0,0,0,(1/14),
                     (6/7),0,0,0,0,0,0,0,(1/7),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (5/6),0,0,0,0,0,0,0,(1/6),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t2")

S3 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(Delta1*haz1$haz[3]+haz2$haz[3]+haz3$haz[3]+haz4$haz[3]+haz5$haz[3]+haz6$haz[3
]+lamdafm7est),Delta1*haz1$haz[3],haz2$haz[3],haz3$haz[3],haz4$haz[3],haz5$haz
[3],haz6$haz[3],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t3")

S4 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(Delta1*haz1$haz[4]+haz2$haz[4]+haz3$haz[4]+haz4$haz[4]+haz5$haz[4]+haz6$haz[4
]+lamdafm7est),Delta1*haz1$haz[4],haz2$haz[4],haz3$haz[4],haz4$haz[4],haz5$haz
[4],haz6$haz[4],lamdafm7est, 0,
                     (4/5),0,0,0,0,0,0,0,(1/5),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t4")

S5 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(Delta1*haz1$haz[5]+haz2$haz[5]+haz3$haz[5]+haz4$haz[5]+haz5$haz[5]+haz6$haz[5
]+lamdafm7est),Delta1*haz1$haz[5],haz2$haz[5],haz3$haz[5],haz4$haz[5],haz5$haz
[5],haz6$haz[5],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t5")

S6 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(Delta1*haz1$haz[6]+haz2$haz[6]+haz3$haz[6]+haz4$haz[6]+haz5$haz[6]+haz6$haz[6
]+lamdafm7est),Delta1*haz1$haz[6],haz2$haz[6],haz3$haz[6],haz4$haz[6],haz5$haz
[6],haz6$haz[6],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name = 
"state t6")

S7 <- new("markovchain", states = stateNames, transitionMatrix = 
            matrix(c(1-
(Delta1*haz1$haz[7]+haz2$haz[7]+haz3$haz[7]+haz4$haz[7]+haz5$haz[7]+haz6$haz[7
]+lamdafm7est),Delta1*haz1$haz[7],haz2$haz[7],haz3$haz[7],haz4$haz[7],haz5$haz
[7],haz6$haz[7],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name = 
"state t7")

S8 <- new("markovchain", states = stateNames, transitionMatrix = 
            matrix(c(1-
(Delta1*haz1$haz[8]+haz2$haz[8]+haz3$haz[8]+haz4$haz[8]+haz5$haz[8]+haz6$haz[8
]+lamdafm7est),Delta1*haz1$haz[8],haz2$haz[8],haz3$haz[8],haz4$haz[8],haz5$haz
[8],haz6$haz[8],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name = 
"state t8")

S9 <- new("markovchain", states = stateNames, transitionMatrix = 
            matrix(c(1-
(Delta1*haz1$haz[9]+haz2$haz[9]+haz3$haz[9]+haz4$haz[9]+haz5$haz[9]+haz6$haz[9
]+lamdafm7est),Delta1*haz1$haz[9],haz2$haz[9],haz3$haz[9],haz4$haz[9],haz5$haz
[9],haz6$haz[9],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name = 
"state t9")
```

```
mcLCASDnew1 <- new("markovchainList",markovchains =
list(S1,S2,S3,S4,S5,S6,S7,S8,S9),
                   name = "Attritable System Behavior with altered FM1")

#calculating the probability of entering each state for the new FM1 design
using % change of baseline
finalStatet1new1 <- initialStatet1*mcLCASDnew1[[1]]^step
finalStatet2new1 <- finalStatet1new1*mcLCASDnew1[[2]]^step
finalStatet3new1 <- finalStatet2new1*mcLCASDnew1[[3]]^step
finalStatet4new1 <- finalStatet3new1*mcLCASDnew1[[4]]^step
finalStatet5new1 <- finalStatet4new1*mcLCASDnew1[[5]]^step
finalStatet6new1 <- finalStatet5new1*mcLCASDnew1[[6]]^step
finalStatet7new1 <- finalStatet6new1*mcLCASDnew1[[7]]^step
finalStatet8new1 <- finalStatet7new1*mcLCASDnew1[[8]]^step
finalStatet9new1 <- finalStatet8new1*mcLCASDnew1[[9]]^step

#######################
# Purpose: create information for altered FM2 Markov Chain Model (MCM)
# inputs: haz1-haz7
# outputs: recursion(1-9)up, mcLCASDnew2, FinalStatenew2t1-9 because there's
one for each time step.
# Author: Bryan Bentz
#######################

Delta2 = 1.0 #ratio of new hazard rate over old hazard rate
stateNames <- c("Operational", "Electronics Bay", "Fuel
Management","Launcher","Operator",
                "Propulsion","Recovery","Structure","Destroyed")
T1 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[1]+Delta2*haz2$haz[1]+haz3$haz[1]+haz4$haz[1]+haz5$haz[1]+haz6$haz[1
]+lamdafm7est),haz1$haz[1],Delta2*haz2$haz[1],haz3$haz[1],haz4$haz[1],haz5$haz
[1],haz6$haz[1],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (12/13),0,0,0,0,0,0,0,(1/13),
                     .75,0,0,0,0,0,0,0,.25,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     .75,0,0,0,0,0,0,0,.25,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t1")
T2 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[2]+Delta2*haz2$haz[2]+haz3$haz[2]+haz4$haz[2]+haz5$haz[2]+haz6$haz[2
]+lamdafm7est),haz1$haz[2],Delta2*haz2$haz[2],haz3$haz[2],haz4$haz[2],haz5$haz
[2],haz6$haz[2],lamdafm7est, 0,
                     (13/14),0,0,0,0,0,0,0,(1/14),
                     (6/7),0,0,0,0,0,0,0,(1/7),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (5/6),0,0,0,0,0,0,0,(1/6),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t2")

T3 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[3]+Delta2*haz2$haz[3]+haz3$haz[3]+haz4$haz[3]+haz5$haz[3]+haz6$haz[3
]+lamdafm7est),haz1$haz[3],Delta2*haz2$haz[3],haz3$haz[3],haz4$haz[3],haz5$haz
[3],haz6$haz[3],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t3")

T4 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[4]+Delta2*haz2$haz[4]+haz3$haz[4]+haz4$haz[4]+haz5$haz[4]+haz6$haz[4
]+lamdafm7est),haz1$haz[4],Delta2*haz2$haz[4],haz3$haz[4],haz4$haz[4],haz5$haz
[4],haz6$haz[4],lamdafm7est, 0,
                     (4/5),0,0,0,0,0,0,0,(1/5),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t4")

T5 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[5]+Delta2*haz2$haz[5]+haz3$haz[5]+haz4$haz[5]+haz5$haz[5]+haz6$haz[5
]+lamdafm7est),haz1$haz[5],Delta2*haz2$haz[5],haz3$haz[5],haz4$haz[5],haz5$haz
[5],haz6$haz[5],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t5")

T6 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[6]+Delta2*haz2$haz[6]+haz3$haz[6]+haz4$haz[6]+haz5$haz[6]+haz6$haz[6
]+lamdafm7est),haz1$haz[6],Delta2*haz2$haz[6],haz3$haz[6],haz4$haz[6],haz5$haz
[6],haz6$haz[6],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t6")

T7 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[7]+Delta2*haz2$haz[7]+haz3$haz[7]+haz4$haz[7]+haz5$haz[7]+haz6$haz[7
]+lamdafm7est),haz1$haz[7],Delta2*haz2$haz[7],haz3$haz[7],haz4$haz[7],haz5$haz
[7],haz6$haz[7],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t7")

T8 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[8]+Delta2*haz2$haz[8]+haz3$haz[8]+haz4$haz[8]+haz5$haz[8]+haz6$haz[8
]+lamdafm7est),haz1$haz[8],Delta2*haz2$haz[8],haz3$haz[8],haz4$haz[8],haz5$haz
[8],haz6$haz[8],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t8")

T9 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[9]+Delta2*haz2$haz[9]+haz3$haz[9]+haz4$haz[9]+haz5$haz[9]+haz6$haz[9
]+lamdafm7est),haz1$haz[9],Delta2*haz2$haz[9],haz3$haz[9],haz4$haz[9],haz5$haz
[9],haz6$haz[9],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t9")

mcLCASDnew2 <- new("markovchainList",markovchains =
list(T1,T2,T3,T4,T5,T6,T7,T8,T9),
                   name = "Attritable System Behavior with FM2 changed")

#calculating the probability of entering each state for the new FM2 design
using % change of baseline
finalStatet1new2 <- initialStatet1*mcLCASDnew2[[1]]^step
finalStatet2new2 <- finalStatet1new2*mcLCASDnew2[[2]]^step
finalStatet3new2 <- finalStatet2new2*mcLCASDnew2[[3]]^step
finalStatet4new2 <- finalStatet3new2*mcLCASDnew2[[4]]^step
finalStatet5new2 <- finalStatet4new2*mcLCASDnew2[[5]]^step
finalStatet6new2 <- finalStatet5new2*mcLCASDnew2[[6]]^step
finalStatet7new2 <- finalStatet6new2*mcLCASDnew2[[7]]^step
finalStatet8new2 <- finalStatet7new2*mcLCASDnew2[[8]]^step
finalStatet9new2 <- finalStatet8new2*mcLCASDnew2[[9]]^step

#######################
# Purpose: create information for altered FM3 Markov Chain Model (MCM)
# inputs: haz1-haz7
# outputs: recursion(1-9)up, mcLCASDnew3, finalStatenew3t1-9 because there's
one for each time step.
# Author: Bryan Bentz
#######################

Delta3 = 1.0 #ratio of new hazard rate over old hazard rate
stateNames <- c("Operational", "Electronics Bay", "Fuel
Management","Launcher","Operator",
                "Propulsion","Recovery","Structure","Destroyed")
W1 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[1]+haz2$haz[1]+Delta3*haz3$haz[1]+haz4$haz[1]+haz5$haz[1]+haz6$haz[1
]+lamdafm7est),haz1$haz[1],haz2$haz[1],Delta3*haz3$haz[1],haz4$haz[1],haz5$haz
[1],haz6$haz[1],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (12/13),0,0,0,0,0,0,0,(1/13),
                     .75,0,0,0,0,0,0,0,.25,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     .75,0,0,0,0,0,0,0,.25,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t1")
W2 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[2]+haz2$haz[2]+Delta3*haz3$haz[2]+haz4$haz[2]+haz5$haz[2]+haz6$haz[2
]+lamdafm7est),haz1$haz[2],haz2$haz[2],Delta3*haz3$haz[2],haz4$haz[2],haz5$haz
[2],haz6$haz[2],lamdafm7est, 0,
                     (13/14),0,0,0,0,0,0,0,(1/14),
                     (6/7),0,0,0,0,0,0,0,(1/7),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (5/6),0,0,0,0,0,0,0,(1/6),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t2")

W3 <- new("markovchain", states = stateNames, transitionMatrix =
           matrix(c(1-
(haz1$haz[3]+haz2$haz[3]+Delta3*haz3$haz[3]+haz4$haz[3]+haz5$haz[3]+haz6$haz[3
]+lamdafm7est),haz1$haz[3],haz2$haz[3],Delta3*haz3$haz[3],haz4$haz[3],haz5$haz
[3],haz6$haz[3],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t3")

W4 <- new("markovchain", states = stateNames, transitionMatrix =
           matrix(c(1-
(haz1$haz[4]+haz2$haz[4]+Delta3*haz3$haz[4]+haz4$haz[4]+haz5$haz[4]+haz6$haz[4
]+lamdafm7est),haz1$haz[4],haz2$haz[4],Delta3*haz3$haz[4],haz4$haz[4],haz5$haz
[4],haz6$haz[4],lamdafm7est, 0,
                     (4/5),0,0,0,0,0,0,0,(1/5),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t4")

W5 <- new("markovchain", states = stateNames, transitionMatrix =
           matrix(c(1-
(haz1$haz[5]+haz2$haz[5]+Delta3*haz3$haz[5]+haz4$haz[5]+haz5$haz[5]+haz6$haz[5
]+lamdafm7est),haz1$haz[5],haz2$haz[5],Delta3*haz3$haz[5],haz4$haz[5],haz5$haz
[5],haz6$haz[5],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t5")
```

```
W6 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[6]+haz2$haz[6]+Delta3*haz3$haz[6]+haz4$haz[6]+haz5$haz[6]+haz6$haz[6
]+lamdafm7est),haz1$haz[6],haz2$haz[6],Delta3*haz3$haz[6],haz4$haz[6],haz5$haz
[6],haz6$haz[6],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t6")

W7 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[7]+haz2$haz[7]+Delta3*haz3$haz[7]+haz4$haz[7]+haz5$haz[7]+haz6$haz[7
]+lamdafm7est),haz1$haz[7],haz2$haz[7],Delta3*haz3$haz[7],haz4$haz[7],haz5$haz
[7],haz6$haz[7],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t7")

W8 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[8]+haz2$haz[8]+Delta3*haz3$haz[8]+haz4$haz[8]+haz5$haz[8]+haz6$haz[8
]+lamdafm7est),haz1$haz[8],haz2$haz[8],Delta3*haz3$haz[8],haz4$haz[8],haz5$haz
[8],haz6$haz[8],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t8")

W9 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[9]+haz2$haz[9]+Delta3*haz3$haz[9]+haz4$haz[9]+haz5$haz[9]+haz6$haz[9
]+lamdafm7est),haz1$haz[9],haz2$haz[9],Delta3*haz3$haz[9],haz4$haz[9],haz5$haz
[9],haz6$haz[9],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t9")

mcLCASDnew3 <- new("markovchainList",markovchains =
list(W1,W2,W3,W4,W5,W6,W7,W8,W9),
                   name = "Attritable System Behavior with FM3 changed")

#calculating the probability of entering each state for the new FM3 design
using % change of baseline
finalStatet1new3 <- initialStatet1*mcLCASDnew3[[1]]^step
finalStatet2new3 <- finalStatet1new3*mcLCASDnew3[[2]]^step
finalStatet3new3 <- finalStatet2new3*mcLCASDnew3[[3]]^step
finalStatet4new3 <- finalStatet3new3*mcLCASDnew3[[4]]^step
finalStatet5new3 <- finalStatet4new3*mcLCASDnew3[[5]]^step
finalStatet6new3 <- finalStatet5new3*mcLCASDnew3[[6]]^step
finalStatet7new3 <- finalStatet6new3*mcLCASDnew3[[7]]^step
finalStatet8new3 <- finalStatet7new3*mcLCASDnew3[[8]]^step
finalStatet9new3 <- finalStatet8new3*mcLCASDnew3[[9]]^step

#######################
# Purpose: create information for altered FM4 Markov Chain Model (MCM)
# inputs: haz1-haz7
# outputs: recursion(1-9)up, mcLCASDnew4, finalStatenew4t1-9 because there's
one for each time step.
# Author: Bryan Bentz
#######################

Delta4 = 1.0 #ratio of new hazard rate over old hazard rate
stateNames <- c("Operational", "Electronics Bay", "Fuel
Management","Launcher","Operator",
                "Propulsion","Recovery","Structure","Destroyed")
X1 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[1]+haz2$haz[1]+haz3$haz[1]+Delta4*haz4$haz[1]+haz5$haz[1]+haz6$haz[1
]+lamdafm7est),haz1$haz[1],haz2$haz[1],haz3$haz[1],Delta4*haz4$haz[1],haz5$haz
[1],haz6$haz[1],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (12/13),0,0,0,0,0,0,0,(1/13),
                     .75,0,0,0,0,0,0,0,.25,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     .75,0,0,0,0,0,0,0,.25,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t1")
X2 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[2]+haz2$haz[2]+haz3$haz[2]+Delta4*haz4$haz[2]+haz5$haz[2]+haz6$haz[2
]+lamdafm7est),haz1$haz[2],haz2$haz[2],haz3$haz[2],Delta4*haz4$haz[2],haz5$haz
```

```
[2],haz6$haz[2],lamdafm7est, 0,
                      (13/14),0,0,0,0,0,0,0,(1/14),
                      (6/7),0,0,0,0,0,0,0,(1/7),
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                      (5/6),0,0,0,0,0,0,0,(1/6),
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                      0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t2")

X3 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[3]+haz2$haz[3]+haz3$haz[3]+Delta4*haz4$haz[3]+haz5$haz[3]+haz6$haz[3
]+lamdafm7est),haz1$haz[3],haz2$haz[3],haz3$haz[3],Delta4*haz4$haz[3],haz5$haz
[3],haz6$haz[3],lamdafm7est, 0,
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                      (2/3),0,0,0,0,0,0,0,(1/3),
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                      0,0,0,0,0,0,0,0,1,
                      0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t3")

X4 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[4]+haz2$haz[4]+haz3$haz[4]+Delta4*haz4$haz[4]+haz5$haz[4]+haz6$haz[4
]+lamdafm7est),haz1$haz[4],haz2$haz[4],haz3$haz[4],Delta4*haz4$haz[4],haz5$haz
[4],haz6$haz[4],lamdafm7est, 0,
                      (4/5),0,0,0,0,0,0,0,(1/5),
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                      0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t4")

X5 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[5]+haz2$haz[5]+haz3$haz[5]+Delta4*haz4$haz[5]+haz5$haz[5]+haz6$haz[5
]+lamdafm7est),haz1$haz[5],haz2$haz[5],haz3$haz[5],Delta4*haz4$haz[5],haz5$haz
[5],haz6$haz[5],lamdafm7est, 0,
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                      1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t5")

X6 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[6]+haz2$haz[6]+haz3$haz[6]+Delta4*haz4$haz[6]+haz5$haz[6]+haz6$haz[6
]+lamdafm7est),haz1$haz[6],haz2$haz[6],haz3$haz[6],Delta4*haz4$haz[6],haz5$haz
[6],haz6$haz[6],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t6")

X7 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[7]+haz2$haz[7]+haz3$haz[7]+Delta4*haz4$haz[7]+haz5$haz[7]+haz6$haz[7
]+lamdafm7est),haz1$haz[7],haz2$haz[7],haz3$haz[7],Delta4*haz4$haz[7],haz5$haz
[7],haz6$haz[7],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t7")

X8 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[8]+haz2$haz[8]+haz3$haz[8]+Delta4*haz4$haz[8]+haz5$haz[8]+haz6$haz[8
]+lamdafm7est),haz1$haz[8],haz2$haz[8],haz3$haz[8],Delta4*haz4$haz[8],haz5$haz
[8],haz6$haz[8],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t8")

X9 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[9]+haz2$haz[9]+haz3$haz[9]+Delta4*haz4$haz[9]+haz5$haz[9]+haz6$haz[9
]+lamdafm7est),haz1$haz[9],haz2$haz[9],haz3$haz[9],Delta4*haz4$haz[9],haz5$haz
[9],haz6$haz[9],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t9")

mcLCASDnew4 <- new("markovchainList",markovchains =
list(X1,X2,X3,X4,X5,X6,X7,X8,X9),
                   name = "Attritable System Behavior with FM4 changed")

#calculating the probability of entering each state for the new FM4 design
using % change of baseline
finalStatet1new4 <- initialStatet1*mcLCASDnew4[[1]]^step
finalStatet2new4 <- finalStatet1new4*mcLCASDnew4[[2]]^step
finalStatet3new4 <- finalStatet2new4*mcLCASDnew4[[3]]^step
finalStatet4new4 <- finalStatet3new4*mcLCASDnew4[[4]]^step
finalStatet5new4 <- finalStatet4new4*mcLCASDnew4[[5]]^step
finalStatet6new4 <- finalStatet5new4*mcLCASDnew4[[6]]^step
finalStatet7new4 <- finalStatet6new4*mcLCASDnew4[[7]]^step
finalStatet8new4 <- finalStatet7new4*mcLCASDnew4[[8]]^step
finalStatet9new4 <- finalStatet8new4*mcLCASDnew4[[9]]^step

#######################
# Purpose: create information for altered FM5 Markov Chain Model (MCM)
# inputs: haz1-haz7
# outputs: recursion(1-9)up, mcLCASDnew5, finalStatenew5t1-9 because there's
one for each time step.
# Author: Bryan Bentz
#######################

Delta5 = 1.0 #ratio of new hazard rate over old hazard rate
stateNames <- c("Operational", "Electronics Bay", "Fuel
Management","Launcher","Operator",
                "Propulsion","Recovery","Structure","Destroyed")
Y1 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[1]+haz2$haz[1]+haz3$haz[1]+haz4$haz[1]+Delta5*haz5$haz[1]+haz6$haz[1
]+lamdafm7est),haz1$haz[1],haz2$haz[1],haz3$haz[1],haz4$haz[1],Delta5*haz5$haz
[1],haz6$haz[1],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (12/13),0,0,0,0,0,0,0,(1/13),
                     .75,0,0,0,0,0,0,0,.25,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     .75,0,0,0,0,0,0,0,.25,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t1")
Y2 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[2]+haz2$haz[2]+haz3$haz[2]+haz4$haz[2]+Delta5*haz5$haz[2]+haz6$haz[2
]+lamdafm7est),haz1$haz[2],haz2$haz[2],haz3$haz[2],haz4$haz[2],Delta5*haz5$haz
[2],haz6$haz[2],lamdafm7est, 0,
                     (13/14),0,0,0,0,0,0,0,(1/14),
                     (6/7),0,0,0,0,0,0,0,(1/7),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (5/6),0,0,0,0,0,0,0,(1/6),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t2")

Y3 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[3]+haz2$haz[3]+haz3$haz[3]+haz4$haz[3]+Delta5*haz5$haz[3]+haz6$haz[3
]+lamdafm7est),haz1$haz[3],haz2$haz[3],haz3$haz[3],haz4$haz[3],Delta5*haz5$haz
[3],haz6$haz[3],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t3")

Y4 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[4]+haz2$haz[4]+haz3$haz[4]+haz4$haz[4]+Delta5*haz5$haz[4]+haz6$haz[4
]+lamdafm7est),haz1$haz[4],haz2$haz[4],haz3$haz[4],haz4$haz[4],Delta5*haz5$haz
[4],haz6$haz[4],lamdafm7est, 0,
                     (4/5),0,0,0,0,0,0,0,(1/5),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t4")

Y5 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[5]+haz2$haz[5]+haz3$haz[5]+haz4$haz[5]+Delta5*haz5$haz[5]+haz6$haz[5
]+lamdafm7est),haz1$haz[5],haz2$haz[5],haz3$haz[5],haz4$haz[5],Delta5*haz5$haz
[5],haz6$haz[5],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t5")

Y6 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[6]+haz2$haz[6]+haz3$haz[6]+haz4$haz[6]+Delta5*haz5$haz[6]+haz6$haz[6
]+lamdafm7est),haz1$haz[6],haz2$haz[6],haz3$haz[6],haz4$haz[6],Delta5*haz5$haz
[6],haz6$haz[6],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t6")

Y7 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[7]+haz2$haz[7]+haz3$haz[7]+haz4$haz[7]+Delta5*haz5$haz[7]+haz6$haz[7
]+lamdafm7est),haz1$haz[7],haz2$haz[7],haz3$haz[7],haz4$haz[7],Delta5*haz5$haz
[7],haz6$haz[7],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t7")

Y8 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[8]+haz2$haz[8]+haz3$haz[8]+haz4$haz[8]+Delta5*haz5$haz[8]+haz6$haz[8
]+lamdafm7est),haz1$haz[8],haz2$haz[8],haz3$haz[8],haz4$haz[8],Delta5*haz5$haz
[8],haz6$haz[8],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t8")

Y9 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[9]+haz2$haz[9]+haz3$haz[9]+haz4$haz[9]+Delta5*haz5$haz[9]+haz6$haz[9
]+lamdafm7est),haz1$haz[9],haz2$haz[9],haz3$haz[9],haz4$haz[9],Delta5*haz5$haz
[9],haz6$haz[9],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t9")

mcLCASDnew5 <- new("markovchainList",markovchains =
list(Y1,Y2,Y3,Y4,Y5,Y6,Y7,Y8,Y9),
                   name = "Attritable System Behavior with FM5 changed")

#calculating the probability of entering each state for the new FM5 design
using % change of baseline
finalStatet1new5 <- initialStatet1*mcLCASDnew5[[1]]^step
finalStatet2new5 <- finalStatet1new5*mcLCASDnew5[[2]]^step
finalStatet3new5 <- finalStatet2new5*mcLCASDnew5[[3]]^step
finalStatet4new5 <- finalStatet3new5*mcLCASDnew5[[4]]^step
finalStatet5new5 <- finalStatet4new5*mcLCASDnew5[[5]]^step
finalStatet6new5 <- finalStatet5new5*mcLCASDnew5[[6]]^step
finalStatet7new5 <- finalStatet6new5*mcLCASDnew5[[7]]^step
finalStatet8new5 <- finalStatet7new5*mcLCASDnew5[[8]]^step
finalStatet9new5 <- finalStatet8new5*mcLCASDnew5[[9]]^step

#######################
# Purpose: create information for altered FM6 Markov Chain Model (MCM)
# inputs: haz1-haz7
# outputs: recursion(1-9)up, mcLCASDnew6, finalStatenew6t1-9 because there's
one for each time step.
# Author: Bryan Bentz
#######################

Delta6 = 1.0 #ratio of new hazard rate over old hazard rate
stateNames <- c("Operational", "Electronics Bay", "Fuel
Management","Launcher","Operator",
                "Propulsion","Recovery","Structure","Destroyed")
Z1 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[1]+haz2$haz[1]+haz3$haz[1]+haz4$haz[1]+haz5$haz[1]+Delta6*haz6$haz[1
]+lamdafm7est),haz1$haz[1],haz2$haz[1],haz3$haz[1],haz4$haz[1],haz5$haz[1],Del
ta6*haz6$haz[1],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (12/13),0,0,0,0,0,0,0,(1/13),
                     .75,0,0,0,0,0,0,0,.25,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     .75,0,0,0,0,0,0,0,.25,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name = 
"state t1")
Z2 <- new("markovchain", states = stateNames, transitionMatrix = 
            matrix(c(1-
(haz1$haz[2]+haz2$haz[2]+haz3$haz[2]+haz4$haz[2]+haz5$haz[2]+Delta6*haz6$haz[2
]+lamdafm7est),haz1$haz[2],haz2$haz[2],haz3$haz[2],haz4$haz[2],haz5$haz[2],Del
ta6*haz6$haz[2],lamdafm7est, 0,
                     (13/14),0,0,0,0,0,0,0,(1/14),
                     (6/7),0,0,0,0,0,0,0,(1/7),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (5/6),0,0,0,0,0,0,0,(1/6),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name = 
"state t2")

Z3 <- new("markovchain", states = stateNames, transitionMatrix = 
            matrix(c(1-
(haz1$haz[3]+haz2$haz[3]+haz3$haz[3]+haz4$haz[3]+haz5$haz[3]+Delta6*haz6$haz[3
]+lamdafm7est),haz1$haz[3],haz2$haz[3],haz3$haz[3],haz4$haz[3],haz5$haz[3],Del
ta6*haz6$haz[3],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name = 
"state t3")

Z4 <- new("markovchain", states = stateNames, transitionMatrix = 
            matrix(c(1-
(haz1$haz[4]+haz2$haz[4]+haz3$haz[4]+haz4$haz[4]+haz5$haz[4]+Delta6*haz6$haz[4
]+lamdafm7est),haz1$haz[4],haz2$haz[4],haz3$haz[4],haz4$haz[4],haz5$haz[4],Del
ta6*haz6$haz[4],lamdafm7est, 0,
                     (4/5),0,0,0,0,0,0,0,(1/5),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name = 
"state t4")

Z5 <- new("markovchain", states = stateNames, transitionMatrix = 
            matrix(c(1-
(haz1$haz[5]+haz2$haz[5]+haz3$haz[5]+haz4$haz[5]+haz5$haz[5]+Delta6*haz6$haz[5
]+lamdafm7est),haz1$haz[5],haz2$haz[5],haz3$haz[5],haz4$haz[5],haz5$haz[5],Del
ta6*haz6$haz[5],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t5")

Z6 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[6]+haz2$haz[6]+haz3$haz[6]+haz4$haz[6]+haz5$haz[6]+Delta6*haz6$haz[6
]+lamdafm7est),haz1$haz[6],haz2$haz[6],haz3$haz[6],haz4$haz[6],haz5$haz[6],Del
ta6*haz6$haz[6],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t6")

Z7 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[7]+haz2$haz[7]+haz3$haz[7]+haz4$haz[7]+haz5$haz[7]+Delta6*haz6$haz[7
]+lamdafm7est),haz1$haz[7],haz2$haz[7],haz3$haz[7],haz4$haz[7],haz5$haz[7],Del
ta6*haz6$haz[7],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t7")

Z8 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[8]+haz2$haz[8]+haz3$haz[8]+haz4$haz[8]+haz5$haz[8]+Delta6*haz6$haz[8
]+lamdafm7est),haz1$haz[8],haz2$haz[8],haz3$haz[8],haz4$haz[8],haz5$haz[8],Del
ta6*haz6$haz[8],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t8")
```

```
Z9 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[9]+haz2$haz[9]+haz3$haz[9]+haz4$haz[9]+haz5$haz[9]+Delta6*haz6$haz[9
]+lamdafm7est),haz1$haz[9],haz2$haz[9],haz3$haz[9],haz4$haz[9],haz5$haz[9],Del
ta6*haz6$haz[9],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t9")

mcLCASDnew6 <- new("markovchainList",markovchains =
list(Z1,Z2,Z3,Z4,Z5,Z6,Z7,Z8,Z9),
                   name = "Attritable System Behavior with FM6 changed")

#calculating the probability of entering each state for the new FM6 design
using % change of baseline
finalStatet1new6 <- initialStatet1*mcLCASDnew6[[1]]^step
finalStatet2new6 <- finalStatet1new6*mcLCASDnew6[[2]]^step
finalStatet3new6 <- finalStatet2new6*mcLCASDnew6[[3]]^step
finalStatet4new6 <- finalStatet3new6*mcLCASDnew6[[4]]^step
finalStatet5new6 <- finalStatet4new6*mcLCASDnew6[[5]]^step
finalStatet6new6 <- finalStatet5new6*mcLCASDnew6[[6]]^step
finalStatet7new6 <- finalStatet6new6*mcLCASDnew6[[7]]^step
finalStatet8new6 <- finalStatet7new6*mcLCASDnew6[[8]]^step
finalStatet9new6 <- finalStatet8new6*mcLCASDnew6[[9]]^step

#######################
# Purpose: create information for altered FM7 Markov Chain Model (MCM)
# inputs: haz1-haz7
# outputs: recursion(1-9)up, mcLCASDnew7, finalStatenew7t1-9 because there's
one for each time step.
# Author: Bryan Bentz
#######################

Delta7 = 1.0 #ratio of new hazard rate over old hazard rate
stateNames <- c("Operational", "Electronics Bay", "Fuel
Management","Launcher","Operator",
                "Propulsion","Recovery","Structure","Destroyed")
A1 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[1]+haz2$haz[1]+haz3$haz[1]+haz4$haz[1]+haz5$haz[1]+haz6$haz[1]+Delta
7*lamdafm7est),haz1$haz[1],haz2$haz[1],haz3$haz[1],haz4$haz[1],haz5$haz[1],haz
6$haz[1],Delta7*lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (12/13),0,0,0,0,0,0,0,(1/13),
                     .75,0,0,0,0,0,0,0,.25,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     .75,0,0,0,0,0,0,0,.25,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name = 
"state t1")
A2 <- new("markovchain", states = stateNames, transitionMatrix = 
            matrix(c(1-
(haz1$haz[2]+haz2$haz[2]+haz3$haz[2]+haz4$haz[2]+haz5$haz[2]+haz6$haz[2]+Delta
7*lamdafm7est),haz1$haz[2],haz2$haz[2],haz3$haz[2],haz4$haz[2],haz5$haz[2],haz
6$haz[2],Delta7*lamdafm7est, 0,
                     (13/14),0,0,0,0,0,0,0,(1/14),
                     (6/7),0,0,0,0,0,0,0,(1/7),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (5/6),0,0,0,0,0,0,0,(1/6),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name = 
"state t2")

A3 <- new("markovchain", states = stateNames, transitionMatrix = 
            matrix(c(1-
(haz1$haz[3]+haz2$haz[3]+haz3$haz[3]+haz4$haz[3]+haz5$haz[3]+haz6$haz[3]+Delta
7*lamdafm7est),haz1$haz[3],haz2$haz[3],haz3$haz[3],haz4$haz[3],haz5$haz[3],haz
6$haz[3],Delta7*lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name = 
"state t3")

A4 <- new("markovchain", states = stateNames, transitionMatrix = 
            matrix(c(1-
(haz1$haz[4]+haz2$haz[4]+haz3$haz[4]+haz4$haz[4]+haz5$haz[4]+haz6$haz[4]+Delta
7*lamdafm7est),haz1$haz[4],haz2$haz[4],haz3$haz[4],haz4$haz[4],haz5$haz[4],haz
6$haz[4],Delta7*lamdafm7est, 0,
                     (4/5),0,0,0,0,0,0,0,(1/5),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name = 
"state t4")

A5 <- new("markovchain", states = stateNames, transitionMatrix = 
            matrix(c(1-
(haz1$haz[5]+haz2$haz[5]+haz3$haz[5]+haz4$haz[5]+haz5$haz[5]+haz6$haz[5]+Delta
7*lamdafm7est),haz1$haz[5],haz2$haz[5],haz3$haz[5],haz4$haz[5],haz5$haz[5],haz
6$haz[5],Delta7*lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t5")

A6 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[6]+haz2$haz[6]+haz3$haz[6]+haz4$haz[6]+haz5$haz[6]+haz6$haz[6]+Delta
7*lamdafm7est),haz1$haz[6],haz2$haz[6],haz3$haz[6],haz4$haz[6],haz5$haz[6],haz
6$haz[6],Delta7*lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t6")

A7 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[7]+haz2$haz[7]+haz3$haz[7]+haz4$haz[7]+haz5$haz[7]+haz6$haz[7]+Delta
7*lamdafm7est),haz1$haz[7],haz2$haz[7],haz3$haz[7],haz4$haz[7],haz5$haz[7],haz
6$haz[7],Delta7*lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t7")

A8 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[8]+haz2$haz[8]+haz3$haz[8]+haz4$haz[8]+haz5$haz[8]+haz6$haz[8]+Delta
7*lamdafm7est),haz1$haz[8],haz2$haz[8],haz3$haz[8],haz4$haz[8],haz5$haz[8],haz
6$haz[8],Delta7*lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t8")

A9 <- new("markovchain", states = stateNames, transitionMatrix =
            matrix(c(1-
(haz1$haz[9]+haz2$haz[9]+haz3$haz[9]+haz4$haz[9]+haz5$haz[9]+haz6$haz[9]+Delta
7*lamdafm7est),haz1$haz[9],haz2$haz[9],haz3$haz[9],haz4$haz[9],haz5$haz[9],haz
6$haz[9],Delta7*lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t9")

mcLCASDnew7 <- new("markovchainList",markovchains =
list(A1,A2,A3,A4,A5,A6,A7,A8,A9),
                   name = "Attritable System Behavior with FM7 changed")

#calculating the probability of entering each state for the new FM7 design
using % change of baseline
finalStatet1new7 <- initialStatet1*mcLCASDnew7[[1]]^step
finalStatet2new7 <- finalStatet1new7*mcLCASDnew7[[2]]^step
finalStatet3new7 <- finalStatet2new7*mcLCASDnew7[[3]]^step
finalStatet4new7 <- finalStatet3new7*mcLCASDnew7[[4]]^step
finalStatet5new7 <- finalStatet4new7*mcLCASDnew7[[5]]^step
finalStatet6new7 <- finalStatet5new7*mcLCASDnew7[[6]]^step
finalStatet7new7 <- finalStatet6new7*mcLCASDnew7[[7]]^step
finalStatet8new7 <- finalStatet7new7*mcLCASDnew7[[8]]^step
finalStatet9new7 <- finalStatet8new7*mcLCASDnew7[[9]]^step

#######################
# Purpose: Calculate S(t) of baseline design with 95% confidence intervals
# inputs: finalStatett1-9,finalState(up&low)t1-9,finalStatenew(i)t1-9
# outputs: yest,ylow,yup,ynew[1-7], vectors of the S(t) at each given time
step for each Markov Chain
#           List model
# Author: Bryan Bentz
#######################

yest <- c((1-finalStatet1[1,9]),(1-finalStatet2[1,9]),(1-
finalStatet3[1,9]),(1-finalStatet4[1,9]),
          (1-finalStatet5[1,9]),(1-finalStatet6[1,9]),(1-
finalStatet7[1,9]),(1-finalStatet8[1,9]),
          (1-finalStatet9[1,9]),(1-finalStatet9[1,9]))
ylow <- c((1-finalStatet1low[1,9]),(1-finalStatet2low[1,9]),(1-
finalStatet3low[1,9]),(1-finalStatet4low[1,9]),
          (1-finalStatet5low[1,9]),(1-finalStatet6low[1,9]),(1-
finalStatet7low[1,9]),(1-finalStatet8low[1,9]),
          (1-finalStatet9low[1,9]),(1-finalStatet9low[1,9]))
yup <- c((1-finalStatet1up[1,9]),(1-finalStatet2up[1,9]),(1-
finalStatet3up[1,9]),(1-finalStatet4up[1,9]),
         (1-finalStatet5up[1,9]),(1-finalStatet6up[1,9]),(1-
finalStatet7up[1,9]),(1-finalStatet8up[1,9]),
         (1-finalStatet9up[1,9]),(1-finalStatet9up[1,9]))
ynew1 <- c((1-finalStatet1new1[1,9]),(1-finalStatet2new1[1,9]),(1-
finalStatet3new1[1,9]),(1-finalStatet4new1[1,9]),
           (1-finalStatet5new1[1,9]),(1-finalStatet6new1[1,9]),(1-
finalStatet7new1[1,9]),(1-finalStatet8new1[1,9]),
           (1-finalStatet9new1[1,9]),(1-finalStatet9new1[1,9]))
ynew2 <- c((1-finalStatet1new2[1,9]),(1-finalStatet2new2[1,9]),(1-
finalStatet3new2[1,9]),(1-finalStatet4new2[1,9]),
           (1-finalStatet5new2[1,9]),(1-finalStatet6new2[1,9]),(1-
finalStatet7new2[1,9]),(1-finalStatet8new2[1,9]),
           (1-finalStatet9new2[1,9]),(1-finalStatet9new2[1,9]))
ynew3 <- c((1-finalStatet1new3[1,9]),(1-finalStatet2new3[1,9]),(1-
finalStatet3new3[1,9]),(1-finalStatet4new3[1,9]),
           (1-finalStatet5new3[1,9]),(1-finalStatet6new3[1,9]),(1-
finalStatet7new3[1,9]),(1-finalStatet8new3[1,9]),
           (1-finalStatet9new3[1,9]),(1-finalStatet9new3[1,9]))
ynew4 <- c((1-finalStatet1new4[1,9]),(1-finalStatet2new4[1,9]),(1-
finalStatet3new4[1,9]),(1-finalStatet4new4[1,9]),
           (1-finalStatet5new4[1,9]),(1-finalStatet6new4[1,9]),(1-
finalStatet7new4[1,9]),(1-finalStatet8new4[1,9]),
           (1-finalStatet9new4[1,9]),(1-finalStatet9new4[1,9]))
ynew5 <- c((1-finalStatet1new5[1,9]),(1-finalStatet2new5[1,9]),(1-
finalStatet3new5[1,9]),(1-finalStatet4new5[1,9]),
           (1-finalStatet5new5[1,9]),(1-finalStatet6new5[1,9]),(1-
finalStatet7new5[1,9]),(1-finalStatet8new5[1,9]),
           (1-finalStatet9new5[1,9]),(1-finalStatet9new5[1,9]))
ynew6 <- c((1-finalStatet1new6[1,9]),(1-finalStatet2new6[1,9]),(1-
finalStatet3new6[1,9]),(1-finalStatet4new6[1,9]),
           (1-finalStatet5new6[1,9]),(1-finalStatet6new6[1,9]),(1-
finalStatet7new6[1,9]),(1-finalStatet8new6[1,9]),
           (1-finalStatet9new6[1,9]),(1-finalStatet9new6[1,9]))
ynew7 <- c((1-finalStatet1new7[1,9]),(1-finalStatet2new7[1,9]),(1-
finalStatet3new7[1,9]),(1-finalStatet4new7[1,9]),
           (1-finalStatet5new7[1,9]),(1-finalStatet6new7[1,9]),(1-
finalStatet7new7[1,9]),(1-finalStatet8new7[1,9]),
           (1-finalStatet9new7[1,9]),(1-finalStatet9new7[1,9]))

#######################
# Purpose: Plot S(t) of designs with baseline 95% confidence intervals
# inputs: mcLCASD - a markovchainList, initialState, steps
# outputs: plot of S(t) over t for every altered hazard rate. Figures 30-35
# Author: Bryan Bentz
#######################

##baseline design###
x=c(0,1,2,3,4,5,6,7,8,9)
par(col='black')
ylower=.965
```

```
plot(x=x, y=yest,ylim=c(ylower,1),main="Survival Function of Baseline System",
     xlab='Sortie',ylab='Prob of Survival',xlim=c(0,9),axes=F, pch = 19)
xticks <- seq(0, 9, 1)
yticks <- seq(0.965,1,0.005)
axis(1, at = xticks, labels = xticks, las=1, tck=-0.01)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(x=x, y=ylow,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 25)
par(new=TRUE)
plot(x=x, y=yup,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 24)
```

```
par(new=FALSE)

##fm1 hazard rate change##
x=c(0,1,2,3,4,5,6,7,8,9)
par(col='black')
ylower=.965
plot(x=x, y=yest,ylim=c(ylower,1),main="Survival Function of FM1 Altered
System",
     xlab='Sortie',ylab='Prob of Survival',xlim=c(0,9),axes=F, pch = 19)
xticks <- seq(0, 9, 1)
yticks <- seq(0.965,1,0.005)
axis(1, at = xticks, labels = xticks, las=1, tck=-0.01)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(x=x, y=ylow,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 25)
par(new=TRUE)
plot(x=x, y=yup,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 24)
par(new=TRUE)
plot(x=x, y=ynew1,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, col = 'red', pch
= 19)
```

```
par(new=FALSE)

##fm2 hazard rate change##etc...
x=c(0,1,2,3,4,5,6,7,8,9)
par(col='black')
ylower=.965
plot(x=x, y=yest,ylim=c(ylower,1),main="Survival Function of FM2 Altered
System",
     xlab='Sortie',ylab='Prob of Survival',xlim=c(0,9),axes=F, pch = 19)
xticks <- seq(0, 9, 1)
yticks <- seq(0.965,1,0.005)
axis(1, at = xticks, labels = xticks, las=1, tck=-0.01)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(x=x, y=ylow,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 25)
par(new=TRUE)
plot(x=x, y=yup,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 24)
par(new=TRUE)
```

```
plot(x=x, y=ynew2,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, col = 'red', pch
= 19)
```

```
par(new=FALSE)

########################

##fm3 hazard rate change##etc...
x=c(0,1,2,3,4,5,6,7,8,9)
par(col='black')
ylower=.965
plot(x=x, y=yest,ylim=c(ylower,1),main="Survival Function of FM3 Altered
System",
     xlab='Sortie',ylab='Prob of Survival',xlim=c(0,9),axes=F, pch = 19)
xticks <- seq(0, 9, 1)
yticks <- seq(0.965,1,0.005)
axis(1, at = xticks, labels = xticks, las=1, tck=-0.01)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(x=x, y=ylow,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 25)
par(new=TRUE)
plot(x=x, y=yup,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 24)
par(new=TRUE)
plot(x=x, y=ynew3,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, col = 'red', pch
= 19)
```

```
par(new=FALSE)

########################

##fm4 hazard rate change##etc...
x=c(0,1,2,3,4,5,6,7,8,9)
par(col='black')
ylower=.965
plot(x=x, y=yest,ylim=c(ylower,1),main="Survival Function of FM4 Altered
System",
     xlab='Sortie',ylab='Prob of Survival',xlim=c(0,9),axes=F, pch = 19)
xticks <- seq(0, 9, 1)
yticks <- seq(0.965,1,0.005)
axis(1, at = xticks, labels = xticks, las=1, tck=-0.01)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(x=x, y=ylow,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 25)
par(new=TRUE)
plot(x=x, y=yup,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 24)
par(new=TRUE)
plot(x=x, y=ynew4,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, col = 'red', pch
= 19)
```

```
par(new=FALSE)

########################

##fm5 hazard rate change##etc...
x=c(0,1,2,3,4,5,6,7,8,9)
par(col='black')
ylower=.965
plot(x=x, y=yest,ylim=c(ylower,1),main="Survival Function of FM5 Altered
System",
     xlab='Sortie',ylab='Prob of Survival',xlim=c(0,9),axes=F, pch = 19)
xticks <- seq(0, 9, 1)
yticks <- seq(0.965,1,0.005)
axis(1, at = xticks, labels = xticks, las=1, tck=-0.01)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(x=x, y=ylow,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 25)
par(new=TRUE)
plot(x=x, y=yup,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 24)
par(new=TRUE)
plot(x=x, y=ynew5,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, col = 'red', pch
= 19)
```

```
par(new=FALSE)

########################

##fm6 hazard rate change##etc...
x=c(0,1,2,3,4,5,6,7,8,9)
par(col='black')
ylower=.965
plot(x=x, y=yest,ylim=c(ylower,1),main="Survival Function of FM6 Altered
System",
     xlab='Sortie',ylab='Prob of Survival',xlim=c(0,9),axes=F, pch = 19)
xticks <- seq(0, 9, 1)
yticks <- seq(0.965,1,0.005)
axis(1, at = xticks, labels = xticks, las=1, tck=-0.01)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(x=x, y=ylow,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 25)
par(new=TRUE)
plot(x=x, y=yup,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 24)
par(new=TRUE)
plot(x=x, y=ynew6,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, col = 'red', pch
= 19)
```

```
par(new=FALSE)

########################
```

```
##fm7 hazard rate change##etc...
x=c(0,1,2,3,4,5,6,7,8,9)
par(col='black')
ylower=.965
plot(x=x, y=yest,ylim=c(ylower,1),main="Survival Function of FM7 Altered
System",
     xlab='Sortie',ylab='Prob of Survival',xlim=c(0,9),axes=F, pch = 19)
xticks <- seq(0, 9, 1)
yticks <- seq(0.965,1,0.005)
axis(1, at = xticks, labels = xticks, las=1, tck=-0.01)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(x=x, y=ylow,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 25)
par(new=TRUE)
plot(x=x, y=yup,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 24)
par(new=TRUE)
plot(x=x, y=ynew7,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, col = 'red', pch
= 19)
```

```
par(new=FALSE)

#######################
# Purpose: Calculate the percentage changein S(t) of after the 9th sortie.
# inputs: yest[1-9], ynew(i)[1-10]
# outputs: maxStdeltafm1-7, this is used to create Figure 36
# Author: Bryan Bentz
#######################

Stdeltafm1 <- c(((yest[10]-ynew1[10])/yest[10])*100)

Stdeltafm2 <- c(((yest[10]-ynew2[10])/yest[10])*100)

Stdeltafm3 <- c(((yest[10]-ynew3[10])/yest[10])*100)

Stdeltafm4 <- c(((yest[10]-ynew4[10])/yest[10])*100)

Stdeltafm5 <- c(((yest[10]-ynew5[10])/yest[10])*100)

Stdeltafm6 <- c(((yest[10]-ynew6[10])/yest[10])*100)

Stdeltafm7 <- c(((yest[10]-ynew7[10])/yest[10])*100)

maxStdeltafm1 <- max(abs(Stdeltafm1))
maxStdeltafm1

## [1] 0

maxStdeltafm2 <- max(abs(Stdeltafm2))
maxStdeltafm2

## [1] 0
```

```
maxStdeltafm3 <- max(abs(Stdeltafm3))
maxStdeltafm3

## [1] 0

maxStdeltafm4 <- max(abs(Stdeltafm4))
maxStdeltafm4

## [1] 0

maxStdeltafm5 <- max(abs(Stdeltafm5))
maxStdeltafm5

## [1] 0

maxStdeltafm6 <- max(abs(Stdeltafm6))
maxStdeltafm6

## [1] 0

maxStdeltafm7 <- max(abs(Stdeltafm7))
maxStdeltafm7

## [1] 0

#######################
# Purpose: create information for an unrepairable FM5 Markov Chain Model (MCM)
# inputs: haz1-haz7
# outputs: recursion(1-9)up, mcLCASDnonrep5, finalStatenonrep5t1-9 because
there's one for each time step.
# Author: Bryan Bentz
#######################

stateNames <- c("Operational", "Electronics Bay", "Fuel
Management","Launcher","Operator",
                "Propulsion","Recovery","Structure","Destroyed")
U1 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion1,haz1$haz[1],haz2$haz[1],haz3$haz[1],haz4$haz[1],haz5$haz[1
],haz6$haz[1],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (12/13),0,0,0,0,0,0,0,(1/13),
                     .75,0,0,0,0,0,0,0,.25,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1,
                     .75,0,0,0,0,0,0,0,.25,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t1")
U2 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion2,haz1$haz[2],haz2$haz[2],haz3$haz[2],haz4$haz[2],haz5$haz[2
],haz6$haz[2],lamdafm7est, 0,
                     (13/14),0,0,0,0,0,0,0,(1/14),
                     (6/7),0,0,0,0,0,0,0,(1/7),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t2")

U3 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion3,haz1$haz[3],haz2$haz[3],haz3$haz[3],haz4$haz[3],haz5$haz[3
],haz6$haz[3],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t3")

U4 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion4,haz1$haz[4],haz2$haz[4],haz3$haz[4],haz4$haz[4],haz5$haz[4
],haz6$haz[4],lamdafm7est, 0,
                     (4/5),0,0,0,0,0,0,0,(1/5),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t4")

U5 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion5,haz1$haz[5],haz2$haz[5],haz3$haz[5],haz4$haz[5],haz5$haz[5
],haz6$haz[5],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t5")

U6 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion6,haz1$haz[6],haz2$haz[6],haz3$haz[6],haz4$haz[6],haz5$haz[6
],haz6$haz[6],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t6")

U7 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion7,haz1$haz[7],haz2$haz[7],haz3$haz[7],haz4$haz[7],haz5$haz[7
],haz6$haz[7],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t7")

U8 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion8,haz1$haz[8],haz2$haz[8],haz3$haz[8],haz4$haz[8],haz5$haz[8
],haz6$haz[8],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t8")

U9 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion9,haz1$haz[9],haz2$haz[9],haz3$haz[9],haz4$haz[9],haz5$haz[9
],haz6$haz[9],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t9")
```

```
mcLCASDnonrep5 <- new("markovchainList",markovchains =
list(U1,U2,U3,U4,U5,U6,U7,U8,U9),
                      name = "Attritable System Behavior with FM5 not
repairable")

finalStatet1nonrep5 <- initialStatet1*mcLCASDnonrep5[[1]]^step
finalStatet2nonrep5 <- finalStatet1nonrep5*mcLCASDnonrep5[[2]]^step
finalStatet3nonrep5 <- finalStatet2nonrep5*mcLCASDnonrep5[[3]]^step
finalStatet4nonrep5 <- finalStatet3nonrep5*mcLCASDnonrep5[[4]]^step
finalStatet5nonrep5 <- finalStatet4nonrep5*mcLCASDnonrep5[[5]]^step
finalStatet6nonrep5 <- finalStatet5nonrep5*mcLCASDnonrep5[[6]]^step
finalStatet7nonrep5 <- finalStatet6nonrep5*mcLCASDnonrep5[[7]]^step
finalStatet8nonrep5 <- finalStatet7nonrep5*mcLCASDnonrep5[[8]]^step
finalStatet9nonrep5 <- finalStatet8nonrep5*mcLCASDnonrep5[[9]]^step

ynonrep5 <- c((1-finalStatet1nonrep5[1,9]),(1-finalStatet2nonrep5[1,9]),(1-
finalStatet3nonrep5[1,9]),(1-finalStatet4nonrep5[1,9]),
              (1-finalStatet5nonrep5[1,9]),(1-finalStatet6nonrep5[1,9]),(1-
finalStatet7nonrep5[1,9]),(1-finalStatet8nonrep5[1,9]),
              (1-finalStatet9nonrep5[1,9]),(1-finalStatet9nonrep5[1,9]))


##fm5 non-repairable change##etc...
x=c(0,1,2,3,4,5,6,7,8,9)
par(col='black')
ylower=.95
plot(x=x, y=yest,ylim=c(ylower,1),main="S(t) without Propulsion Repair",
     xlab='sortie',ylab='Prob of Survival',xlim=c(0,9),axes=F, pch = 19)
xticks <- seq(0, 9, 1)
yticks <- seq(0.95,1,0.01)
axis(1, at = xticks, labels = xticks, las=1, tck=-0.01)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(x=x, y=ylow,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 25)
par(new=TRUE)
plot(x=x, y=yup,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 24)
par(new=TRUE)
plot(x=x, y=ynonrep5,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, col = 'blue',
pch = 15)
```

```
par(new=FALSE)

#######################
# Purpose: create information for an unrepairable FM7 Markov Chain Model (MCM)
# inputs: haz1-haz7
# outputs: recursion(1-9)up, mcLCASDnonrep7, finalStatenonrep7t1-9 because
there's one for each time step.
# Author: Bryan Bentz
#######################

stateNames <- c("Operational", "Electronics Bay", "Fuel
Management","Launcher","Operator",
                "Propulsion","Recovery","Structure","Destroyed")
V1 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion1,haz1$haz[1],haz2$haz[1],haz3$haz[1],haz4$haz[1],haz5$haz[1
],haz6$haz[1],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (12/13),0,0,0,0,0,0,0,(1/13),
                     .75,0,0,0,0,0,0,0,.25,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     .75,0,0,0,0,0,0,0,.25,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t1")
V2 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion2,haz1$haz[2],haz2$haz[2],haz3$haz[2],haz4$haz[2],haz5$haz[2
],haz6$haz[2],lamdafm7est, 0,
                     (13/14),0,0,0,0,0,0,0,(1/14),
                     (6/7),0,0,0,0,0,0,0,(1/7),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (5/6),0,0,0,0,0,0,0,(1/6),
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t2")

V3 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion3,haz1$haz[3],haz2$haz[3],haz3$haz[3],haz4$haz[3],haz5$haz[3
],haz6$haz[3],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t3")

V4 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion4,haz1$haz[4],haz2$haz[4],haz3$haz[4],haz4$haz[4],haz5$haz[4
],haz6$haz[4],lamdafm7est, 0,
                     (4/5),0,0,0,0,0,0,0,(1/5),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t4")

V5 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion5,haz1$haz[5],haz2$haz[5],haz3$haz[5],haz4$haz[5],haz5$haz[5
],haz6$haz[5],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t5")

V6 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion6,haz1$haz[6],haz2$haz[6],haz3$haz[6],haz4$haz[6],haz5$haz[6
],haz6$haz[6],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     (2/3),0,0,0,0,0,0,0,(1/3),
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t6")

V7 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion7,haz1$haz[7],haz2$haz[7],haz3$haz[7],haz4$haz[7],haz5$haz[7
],haz6$haz[7],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t7")

V8 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion8,haz1$haz[8],haz2$haz[8],haz3$haz[8],haz4$haz[8],haz5$haz[8
],haz6$haz[8],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t8")

V9 <- new("markovchain", states = stateNames, transitionMatrix =

matrix(c(recursion9,haz1$haz[9],haz2$haz[9],haz3$haz[9],haz4$haz[9],haz5$haz[9
],haz6$haz[9],lamdafm7est, 0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     1,0,0,0,0,0,0,0,0,
                     0,0,0,0,0,0,0,0,1,
                     0,0,0,0,0,0,0,0,1), byrow = TRUE, nrow = 9), name =
"state t9")

mcLCASDnonrep7 <- new("markovchainList",markovchains =
list(V1,V2,V3,V4,V5,V6,V7,V8,V9),
                      name = "System Behavior with FM7 not repairable")

finalStatet1nonrep7 <- initialStatet1*mcLCASDnonrep7[[1]]^step
finalStatet2nonrep7 <- finalStatet1nonrep7*mcLCASDnonrep7[[2]]^step
finalStatet3nonrep7 <- finalStatet2nonrep7*mcLCASDnonrep7[[3]]^step
finalStatet4nonrep7 <- finalStatet3nonrep7*mcLCASDnonrep7[[4]]^step
finalStatet5nonrep7 <- finalStatet4nonrep7*mcLCASDnonrep7[[5]]^step
finalStatet6nonrep7 <- finalStatet5nonrep7*mcLCASDnonrep7[[6]]^step
finalStatet7nonrep7 <- finalStatet6nonrep7*mcLCASDnonrep7[[7]]^step
finalStatet8nonrep7 <- finalStatet7nonrep7*mcLCASDnonrep7[[8]]^step
finalStatet9nonrep7 <- finalStatet8nonrep7*mcLCASDnonrep7[[9]]^step

ynonrep7 <- c((1-finalStatet1nonrep7[1,9]),(1-finalStatet2nonrep7[1,9]),(1-
finalStatet3nonrep7[1,9]),(1-finalStatet4nonrep7[1,9]),
              (1-finalStatet5nonrep7[1,9]),(1-finalStatet6nonrep7[1,9]),(1-
finalStatet7nonrep7[1,9]),(1-finalStatet8nonrep7[1,9]),
              (1-finalStatet9nonrep7[1,9]),(1-finalStatet9nonrep7[1,9]-
(finalStatet8nonrep7[1,9]-finalStatet7nonrep7[1,9])))


##fm7 non-repairable change##etc...
x=c(0,1,2,3,4,5,6,7,8,9)
par(col='black')
ylower=.95
plot(x=x, y=yest,ylim=c(ylower,1),main="S(t) without Structural Repair",
     xlab='sortie',ylab='Prob of Survival',xlim=c(0,9),axes=F, pch = 19)
xticks <- seq(0, 9, 1)
yticks <- seq(0.95,1,0.01)
axis(1, at = xticks, labels = xticks, las=1, tck=-0.01)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
```

```
plot(x=x, y=ylow,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 25)
par(new=TRUE)
plot(x=x, y=yup,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, pch = 24)
par(new=TRUE)
plot(x=x, y=ynonrep7,ylim=c(ylower,1),ann=F,xlim=c(0,9),axes=F, col = 'green',
pch = 18)
```

```
par(new=FALSE)

############################################
######Absolute Cost Risk Calculations#######
############################################

#######################
# Purpose: Data input of example subsystem costs of acquisition, repair, regen
costs.  See Table 4 in Thesis.
# inputs: N/A
# outputs: Regeneration Cost, Average Per Unit Flyaway Costs, Absolute Costs
of each Markov Chain Transition
# Author: Bryan Bentz
#######################

#example subsystem costs of based on LRC to LAC ratio of buy of 100 UAVs.  See
Table 4

fm1cost1 <- 1000000
fm1cost2 <- 1000000 #Assessment of FM1 design alternative cost
fm2cost <- 50000
fm3cost <- 540000
fm4cost <- 0
fm5cost1 <- 600000
fm5cost2 <- 600000 #Assessment of FM5 design alternative cost
fm6cost <- 30000
fm7cost <- 750000
consumablescost <- 36000 #cost of consumables per sortie
laborcost <- 9000 #estimated cost of labor to regenerate

#####Cost to buy ratios based on input from Campanile, RQQC
fm1cbr <- 0.15
fm2cbr <- 0.32
fm3cbr <- 0.12
fm4cbr <- 0
fm5cbr <- 0.22
fm6cbr <- 0.20
fm7cbr <- 0.32

#Regeneration Costs.  See Section 3.4 for explanation

rbrabscost <- (consumablescost+laborcost)

#Average Per Unit Flyaway Cost Calculations
```

```
flyawaycost1 <- fm1cost1+fm2cost+fm5cost1+fm6cost+fm7cost
flyawaycost2fm1 <- fm1cost2+fm2cost+fm5cost1+fm6cost+fm7cost
flyawaycost2fm5 <- fm1cost1+fm2cost+fm5cost2+fm6cost+fm7cost

#Absolute Costs of each Markov Chain transition

fm1abscost1 <- rbrabscost+(fm1cost1*fm1cbr)
fm1abscost2 <- rbrabscost+(fm1cost2*fm1cbr)
fm2abscost <- rbrabscost+(fm2cost*fm2cbr)
fm3abscost <- rbrabscost+(fm3cost*fm3cbr)
fm4abscost <- rbrabscost+(fm4cost*fm4cbr)
fm5abscost1 <- rbrabscost+(fm5cost1*fm5cbr)
fm5abscost2 <- rbrabscost+(fm5cost2*fm5cbr)
fm6abscost <- rbrabscost+(fm6cost*fm6cbr)
fm7abscost <- rbrabscost+(fm7cost*fm7cbr)

#######################
# Purpose: Calculate absolute cost risks at each interval in time for all
design alternatives
# inputs: N/A
# outputs: 7 vectors describing absolute cost risk at each interval in time
for all alternatives.
# Author: Bryan Bentz
# Notes: the cost of repair are nullified for nonrep5 and nonrep7 calculations
#   the new1 and new5 vectors calculate cost risk using the different
acquisition and flyaway costs
#######################

abscostriskt1est <-
(finalStatet1[1,1]*rbrabscost+finalStatet1[1,2]*fm1abscost1+finalStatet1[1,3]*
fm2abscost+finalStatet1[1,4]*fm3abscost
                     +
finalStatet1[1,5]*fm4abscost+finalStatet1[1,6]*fm5abscost1+finalStatet1[1,7]*f
m6abscost+finalStatet1[1,8]*fm7abscost
                     + finalStatet1[1,9]*flyawaycost1)
abscostriskt2est <-
(finalStatet2[1,1]*rbrabscost+finalStatet2[1,2]*fm1abscost1+finalStatet2[1,3]*
fm2abscost+finalStatet2[1,4]*fm3abscost
                     +
finalStatet2[1,5]*fm4abscost+finalStatet2[1,6]*fm5abscost1+finalStatet2[1,7]*f
m6abscost+finalStatet2[1,8]*fm7abscost
                     + finalStatet2[1,9]*flyawaycost1)
abscostriskt3est <-
(finalStatet3[1,1]*rbrabscost+finalStatet3[1,2]*fm1abscost1+finalStatet3[1,3]*
fm2abscost+finalStatet3[1,4]*fm3abscost
                     +
finalStatet3[1,5]*fm4abscost+finalStatet3[1,6]*fm5abscost1+finalStatet3[1,7]*f
m6abscost+finalStatet3[1,8]*fm7abscost
                     + finalStatet3[1,9]*flyawaycost1)
abscostriskt4est <-
(finalStatet4[1,1]*rbrabscost+finalStatet4[1,2]*fm1abscost1+finalStatet4[1,3]*
fm2abscost+finalStatet4[1,4]*fm3abscost
                     +
finalStatet4[1,5]*fm4abscost+finalStatet4[1,6]*fm5abscost1+finalStatet4[1,7]*f
m6abscost+finalStatet4[1,8]*fm7abscost
                     + finalStatet4[1,9]*flyawaycost1)
abscostriskt5est <-
(finalStatet5[1,1]*rbrabscost+finalStatet5[1,2]*fm1abscost1+finalStatet5[1,3]*
fm2abscost+finalStatet5[1,4]*fm3abscost
                     +
finalStatet5[1,5]*fm4abscost+finalStatet5[1,6]*fm5abscost1+finalStatet5[1,7]*f
m6abscost+finalStatet5[1,8]*fm7abscost
                     + finalStatet5[1,9]*flyawaycost1)
abscostriskt6est <-
(finalStatet6[1,1]*rbrabscost+finalStatet6[1,2]*fm1abscost1+finalStatet6[1,3]*
fm2abscost+finalStatet6[1,4]*fm3abscost
                     +
finalStatet6[1,5]*fm4abscost+finalStatet6[1,6]*fm5abscost1+finalStatet6[1,7]*f
m6abscost+finalStatet6[1,8]*fm7abscost
                     + finalStatet6[1,9]*flyawaycost1)
abscostriskt7est <-
(finalStatet7[1,1]*rbrabscost+finalStatet7[1,2]*fm1abscost1+finalStatet7[1,3]*
fm2abscost+finalStatet7[1,4]*fm3abscost
                     +
finalStatet7[1,5]*fm4abscost+finalStatet7[1,6]*fm5abscost1+finalStatet7[1,7]*f
m6abscost+finalStatet7[1,8]*fm7abscost
                     + finalStatet7[1,9]*flyawaycost1)
abscostriskt8est <-
(finalStatet8[1,1]*rbrabscost+finalStatet8[1,2]*fm1abscost1+finalStatet8[1,3]*
fm2abscost+finalStatet8[1,4]*fm3abscost
                     +
finalStatet8[1,5]*fm4abscost+finalStatet8[1,6]*fm5abscost1+finalStatet8[1,7]*f
m6abscost+finalStatet8[1,8]*fm7abscost
                     + finalStatet8[1,9]*flyawaycost1)
abscostriskt9est <-
(finalStatet9[1,1]*rbrabscost+finalStatet9[1,2]*fm1abscost1+finalStatet9[1,3]*
fm2abscost+finalStatet9[1,4]*fm3abscost
                     +
finalStatet9[1,5]*fm4abscost+finalStatet9[1,6]*fm5abscost1+finalStatet9[1,7]*f
m6abscost+finalStatet9[1,8]*fm7abscost
                     + finalStatet9[1,9]*flyawaycost1)
abscostriskest <- c(0,
abscostriskt1est,abscostriskt2est,abscostriskt3est,abscostriskt4est,abscostris
kt5est,abscostriskt6est,abscostriskt7est,
                    abscostriskt8est,abscostriskt9est)

############################

abscostriskt1nonrep5 <-
(finalStatet1nonrep5[1,1]*rbrabscost+finalStatet1nonrep5[1,2]*fm1abscost1+fina
lStatet1nonrep5[1,3]*fm2abscost+finalStatet1nonrep5[1,4]*fm3abscost
                         +
finalStatet1nonrep5[1,5]*fm4abscost+finalStatet1nonrep5[1,6]*0+finalStatet1non
rep5[1,7]*fm6abscost+finalStatet1nonrep5[1,8]*fm7abscost
                         + finalStatet1nonrep5[1,9]*flyawaycost1)
abscostriskt2nonrep5 <-
```

```
(finalStatet2nonrep5[1,1]*rbrabscost+finalStatet2nonrep5[1,2]*fm1abscost1+fina
lStatet2nonrep5[1,3]*fm2abscost+finalStatet2nonrep5[1,4]*fm3abscost
                         +
finalStatet2nonrep5[1,5]*fm4abscost+finalStatet2nonrep5[1,6]*0+finalStatet2non
rep5[1,7]*fm6abscost+finalStatet2nonrep5[1,8]*fm7abscost
                         + finalStatet2nonrep5[1,9]*flyawaycost1)
abscostriskt3nonrep5 <-
(finalStatet3nonrep5[1,1]*rbrabscost+finalStatet3nonrep5[1,2]*fm1abscost1+fina
lStatet3nonrep5[1,3]*fm2abscost+finalStatet3nonrep5[1,4]*fm3abscost
                         +
finalStatet3nonrep5[1,5]*fm4abscost+finalStatet3nonrep5[1,6]*0+finalStatet3non
rep5[1,7]*fm6abscost+finalStatet3nonrep5[1,8]*fm7abscost
                         + finalStatet3nonrep5[1,9]*flyawaycost1)
abscostriskt4nonrep5 <-
(finalStatet4nonrep5[1,1]*rbrabscost+finalStatet4nonrep5[1,2]*fm1abscost1+fina
lStatet4nonrep5[1,3]*fm2abscost+finalStatet4nonrep5[1,4]*fm3abscost
                         +
finalStatet4nonrep5[1,5]*fm4abscost+finalStatet4nonrep5[1,6]*0+finalStatet4non
rep5[1,7]*fm6abscost+finalStatet4nonrep5[1,8]*fm7abscost
                         + finalStatet4nonrep5[1,9]*flyawaycost1)
abscostriskt5nonrep5 <-
(finalStatet5nonrep5[1,1]*rbrabscost+finalStatet5nonrep5[1,2]*fm1abscost1+fina
lStatet5nonrep5[1,3]*fm2abscost+finalStatet5nonrep5[1,4]*fm3abscost
                         +
finalStatet5nonrep5[1,5]*fm4abscost+finalStatet5nonrep5[1,6]*0+finalStatet5non
rep5[1,7]*fm6abscost+finalStatet5nonrep5[1,8]*fm7abscost
                         + finalStatet5nonrep5[1,9]*flyawaycost1)
abscostriskt6nonrep5 <-
(finalStatet6nonrep5[1,1]*rbrabscost+finalStatet6nonrep5[1,2]*fm1abscost1+fina
lStatet6nonrep5[1,3]*fm2abscost+finalStatet6nonrep5[1,4]*fm3abscost
                         +
finalStatet6nonrep5[1,5]*fm4abscost+finalStatet6nonrep5[1,6]*0+finalStatet6non
rep5[1,7]*fm6abscost+finalStatet6nonrep5[1,8]*fm7abscost
                         + finalStatet6nonrep5[1,9]*flyawaycost1)
abscostriskt7nonrep5 <-
(finalStatet7nonrep5[1,1]*rbrabscost+finalStatet7nonrep5[1,2]*fm1abscost1+fina
lStatet7nonrep5[1,3]*fm2abscost+finalStatet7nonrep5[1,4]*fm3abscost
                         +
finalStatet7nonrep5[1,5]*fm4abscost+finalStatet7nonrep5[1,6]*0+finalStatet7non
rep5[1,7]*fm6abscost+finalStatet7nonrep5[1,8]*fm7abscost
                         + finalStatet7nonrep5[1,9]*flyawaycost1)
abscostriskt8nonrep5 <-
(finalStatet8nonrep5[1,1]*rbrabscost+finalStatet8nonrep5[1,2]*fm1abscost1+fina
lStatet8nonrep5[1,3]*fm2abscost+finalStatet8nonrep5[1,4]*fm3abscost
                         +
finalStatet8nonrep5[1,5]*fm4abscost+finalStatet8nonrep5[1,6]*0+finalStatet8non
rep5[1,7]*fm6abscost+finalStatet8nonrep5[1,8]*fm7abscost
                         + finalStatet8nonrep5[1,9]*flyawaycost1)
abscostriskt9nonrep5 <-
(finalStatet9nonrep5[1,1]*rbrabscost+finalStatet9nonrep5[1,2]*fm1abscost1+fina
lStatet9nonrep5[1,3]*fm2abscost+finalStatet9nonrep5[1,4]*fm3abscost
                         +
finalStatet9nonrep5[1,5]*fm4abscost+finalStatet9nonrep5[1,6]*0+finalStatet9non
```

```
rep5[1,7]*fm6abscost+finalStatet9nonrep5[1,8]*fm7abscost
                          + finalStatet9nonrep5[1,9]*flyawaycost1)
abscostrisknonrep5 <- c(0,
abscostriskt1nonrep5,abscostriskt2nonrep5,abscostriskt3nonrep5,abscostriskt4no
nrep5,abscostriskt5nonrep5,abscostriskt6nonrep5,abscostriskt7nonrep5,
                         abscostriskt8nonrep5,abscostriskt9nonrep5)

####################################
abscostriskt1nonrep7 <-
(finalStatet1nonrep7[1,1]*rbrabscost+finalStatet1nonrep7[1,2]*fm1abscost1+fina
lStatet1nonrep7[1,3]*fm2abscost+finalStatet1nonrep7[1,4]*fm3abscost
                          +
finalStatet1nonrep7[1,5]*fm4abscost+finalStatet1nonrep7[1,6]*fm5abscost1+final
Statet1nonrep7[1,7]*fm6abscost+finalStatet1nonrep7[1,8]*0
                          + finalStatet1nonrep7[1,9]*flyawaycost1)
abscostriskt2nonrep7 <-
(finalStatet2nonrep7[1,1]*rbrabscost+finalStatet2nonrep7[1,2]*fm1abscost1+fina
lStatet2nonrep7[1,3]*fm2abscost+finalStatet2nonrep7[1,4]*fm3abscost
                          +
finalStatet2nonrep7[1,5]*fm4abscost+finalStatet2nonrep7[1,6]*fm5abscost1+final
Statet2nonrep7[1,7]*fm6abscost+finalStatet2nonrep7[1,8]*0
                          + finalStatet2nonrep7[1,9]*flyawaycost1)
abscostriskt3nonrep7 <-
(finalStatet3nonrep7[1,1]*rbrabscost+finalStatet3nonrep7[1,2]*fm1abscost1+fina
lStatet3nonrep7[1,3]*fm2abscost+finalStatet3nonrep7[1,4]*fm3abscost
                          +
finalStatet3nonrep7[1,5]*fm4abscost+finalStatet3nonrep7[1,6]*fm5abscost1+final
Statet3nonrep7[1,7]*fm6abscost+finalStatet3nonrep7[1,8]*0
                          + finalStatet3nonrep7[1,9]*flyawaycost1)
abscostriskt4nonrep7 <-
(finalStatet4nonrep7[1,1]*rbrabscost+finalStatet4nonrep7[1,2]*fm1abscost1+fina
lStatet4nonrep7[1,3]*fm2abscost+finalStatet4nonrep7[1,4]*fm3abscost
                          +
finalStatet4nonrep7[1,5]*fm4abscost+finalStatet4nonrep7[1,6]*fm5abscost1+final
Statet4nonrep7[1,7]*fm6abscost+finalStatet4nonrep7[1,8]*0
                          + finalStatet4nonrep7[1,9]*flyawaycost1)
abscostriskt5nonrep7 <-
(finalStatet5nonrep7[1,1]*rbrabscost+finalStatet5nonrep7[1,2]*fm1abscost1+fina
lStatet5nonrep7[1,3]*fm2abscost+finalStatet5nonrep7[1,4]*fm3abscost
                          +
finalStatet5nonrep7[1,5]*fm4abscost+finalStatet5nonrep7[1,6]*fm5abscost1+final
Statet5nonrep7[1,7]*fm6abscost+finalStatet5nonrep7[1,8]*0
                          + finalStatet5nonrep7[1,9]*flyawaycost1)
abscostriskt6nonrep7 <-
(finalStatet6nonrep7[1,1]*rbrabscost+finalStatet6nonrep7[1,2]*fm1abscost1+fina
lStatet6nonrep7[1,3]*fm2abscost+finalStatet6nonrep7[1,4]*fm3abscost
                          +
finalStatet6nonrep7[1,5]*fm4abscost+finalStatet6nonrep7[1,6]*fm5abscost1+final
Statet6nonrep7[1,7]*fm6abscost+finalStatet6nonrep7[1,8]*0
                          + finalStatet6nonrep7[1,9]*flyawaycost1)
abscostriskt7nonrep7 <-
(finalStatet7nonrep7[1,1]*rbrabscost+finalStatet7nonrep7[1,2]*fm1abscost1+fina
lStatet7nonrep7[1,3]*fm2abscost+finalStatet7nonrep7[1,4]*fm3abscost
```

```
                         +
finalStatet7nonrep7[1,5]*fm4abscost+finalStatet7nonrep7[1,6]*fm5abscost1+final
Statet7nonrep7[1,7]*fm6abscost+finalStatet7nonrep7[1,8]*0
                         + finalStatet7nonrep7[1,9]*flyawaycost1)
abscostriskt8nonrep7 <-
(finalStatet8nonrep7[1,1]*rbrabscost+finalStatet8nonrep7[1,2]*fm1abscost1+fina
lStatet8nonrep7[1,3]*fm2abscost+finalStatet8nonrep7[1,4]*fm3abscost
                         +
finalStatet8nonrep7[1,5]*fm4abscost+finalStatet8nonrep7[1,6]*fm5abscost1+final
Statet8nonrep7[1,7]*fm6abscost+finalStatet8nonrep7[1,8]*0
                         + finalStatet8nonrep7[1,9]*flyawaycost1)
abscostriskt9nonrep7 <-
(finalStatet9nonrep7[1,1]*rbrabscost+finalStatet9nonrep7[1,2]*fm1abscost1+fina
lStatet9nonrep7[1,3]*fm2abscost+finalStatet9nonrep7[1,4]*fm3abscost
                         +
finalStatet9nonrep7[1,5]*fm4abscost+finalStatet9nonrep7[1,6]*fm5abscost1+final
Statet9nonrep7[1,7]*fm6abscost+finalStatet9nonrep7[1,8]*0
                         + finalStatet9nonrep7[1,9]*flyawaycost1)
abscostrisknonrep7 <- c(0,
abscostriskt1nonrep7,abscostriskt2nonrep7,abscostriskt3nonrep7,abscostriskt4no
nrep7,abscostriskt5nonrep7,abscostriskt6nonrep7,abscostriskt7nonrep7,
                        abscostriskt8nonrep7,abscostriskt9nonrep7)


###################################

abscostriskt1low <-
(finalStatet1low[1,1]*rbrabscost+finalStatet1low[1,2]*fm1abscost1+finalStatet1
low[1,3]*fm2abscost+finalStatet1low[1,4]*fm3abscost
                     +
finalStatet1low[1,5]*fm4abscost+finalStatet1low[1,6]*fm5abscost1+finalStatet1l
ow[1,7]*fm6abscost+finalStatet1low[1,8]*fm7abscost
                     + finalStatet1low[1,9]*flyawaycost1)
abscostriskt2low <-
(finalStatet2low[1,1]*rbrabscost+finalStatet2low[1,2]*fm1abscost1+finalStatet2
low[1,3]*fm2abscost+finalStatet2low[1,4]*fm3abscost
                     +
finalStatet2low[1,5]*fm4abscost+finalStatet2low[1,6]*fm5abscost1+finalStatet2l
ow[1,7]*fm6abscost+finalStatet2low[1,8]*fm7abscost
                     + finalStatet2low[1,9]*flyawaycost1)
abscostriskt3low <-
(finalStatet3low[1,1]*rbrabscost+finalStatet3low[1,2]*fm1abscost1+finalStatet3
low[1,3]*fm2abscost+finalStatet3low[1,4]*fm3abscost
                     +
finalStatet3low[1,5]*fm4abscost+finalStatet3low[1,6]*fm5abscost1+finalStatet3l
ow[1,7]*fm6abscost+finalStatet3low[1,8]*fm7abscost
                     + finalStatet3low[1,9]*flyawaycost1)
abscostriskt4low <-
(finalStatet4low[1,1]*rbrabscost+finalStatet4low[1,2]*fm1abscost1+finalStatet4
low[1,3]*fm2abscost+finalStatet4low[1,4]*fm3abscost
                     +
finalStatet4low[1,5]*fm4abscost+finalStatet4low[1,6]*fm5abscost1+finalStatet4l
ow[1,7]*fm6abscost+finalStatet4low[1,8]*fm7abscost
```

```
                     + finalStatet4low[1,9]*flyawaycost1)
abscostriskt5low <-
(finalStatet5low[1,1]*rbrabscost+finalStatet5low[1,2]*fm1abscost1+finalStatet5
low[1,3]*fm2abscost+finalStatet5low[1,4]*fm3abscost
                     +
finalStatet5low[1,5]*fm4abscost+finalStatet5low[1,6]*fm5abscost1+finalStatet5l
ow[1,7]*fm6abscost+finalStatet5low[1,8]*fm7abscost
                     + finalStatet5low[1,9]*flyawaycost1)
abscostriskt6low <-
(finalStatet6low[1,1]*rbrabscost+finalStatet6low[1,2]*fm1abscost1+finalStatet6
low[1,3]*fm2abscost+finalStatet6low[1,4]*fm3abscost
                     +
finalStatet6low[1,5]*fm4abscost+finalStatet6low[1,6]*fm5abscost1+finalStatet6l
ow[1,7]*fm6abscost+finalStatet6low[1,8]*fm7abscost
                     + finalStatet6low[1,9]*flyawaycost1)
abscostriskt7low <-
(finalStatet7low[1,1]*rbrabscost+finalStatet7low[1,2]*fm1abscost1+finalStatet7
low[1,3]*fm2abscost+finalStatet7low[1,4]*fm3abscost
                     +
finalStatet7low[1,5]*fm4abscost+finalStatet7low[1,6]*fm5abscost1+finalStatet7l
ow[1,7]*fm6abscost+finalStatet7low[1,8]*fm7abscost
                     + finalStatet7low[1,9]*flyawaycost1)
abscostriskt8low <-
(finalStatet8low[1,1]*rbrabscost+finalStatet8low[1,2]*fm1abscost1+finalStatet8
low[1,3]*fm2abscost+finalStatet8low[1,4]*fm3abscost
                     +
finalStatet8low[1,5]*fm4abscost+finalStatet8low[1,6]*fm5abscost1+finalStatet8l
ow[1,7]*fm6abscost+finalStatet8low[1,8]*fm7abscost
                     + finalStatet8low[1,9]*flyawaycost1)
abscostriskt9low <-
(finalStatet9low[1,1]*rbrabscost+finalStatet9low[1,2]*fm1abscost1+finalStatet9
low[1,3]*fm2abscost+finalStatet9low[1,4]*fm3abscost
                     +
finalStatet9low[1,5]*fm4abscost+finalStatet9low[1,6]*fm5abscost1+finalStatet9l
ow[1,7]*fm6abscost+finalStatet9low[1,8]*fm7abscost
                     + finalStatet9low[1,9]*flyawaycost1)
abscostrisklow <- c(0,
abscostriskt1low,abscostriskt2low,abscostriskt3low,abscostriskt4low,abscostris
kt5low,abscostriskt6low,abscostriskt7low,
                    abscostriskt8low,abscostriskt9low)

####################################

abscostriskt1up <-
(finalStatet1up[1,1]*rbrabscost+finalStatet1up[1,2]*fm1abscost1+finalStatet1[1
,3]*fm2abscost+finalStatet1up[1,4]*fm3abscost
                    +
finalStatet1up[1,5]*fm4abscost+finalStatet1up[1,6]*fm5abscost1+finalStatet1[1,
7]*fm6abscost+finalStatet1up[1,8]*fm7abscost
                    + finalStatet1up[1,9]*flyawaycost1)
abscostriskt2up <-
(finalStatet2up[1,1]*rbrabscost+finalStatet2up[1,2]*fm1abscost1+finalStatet2[1
,3]*fm2abscost+finalStatet2up[1,4]*fm3abscost
```

```
                    +
finalStatet2up[1,5]*fm4abscost+finalStatet2up[1,6]*fm5abscost1+finalStatet2[1,
7]*fm6abscost+finalStatet2up[1,8]*fm7abscost
                    + finalStatet2up[1,9]*flyawaycost1)
abscostriskt3up <-
(finalStatet3up[1,1]*rbrabscost+finalStatet3up[1,2]*fm1abscost1+finalStatet3up
[1,3]*fm2abscost+finalStatet3up[1,4]*fm3abscost
                    +
finalStatet3up[1,5]*fm4abscost+finalStatet3up[1,6]*fm5abscost1+finalStatet3up[
1,7]*fm6abscost+finalStatet3up[1,8]*fm7abscost
                    + finalStatet3up[1,9]*flyawaycost1)
abscostriskt4up <-
(finalStatet4up[1,1]*rbrabscost+finalStatet4up[1,2]*fm1abscost1+finalStatet4up
[1,3]*fm2abscost+finalStatet4up[1,4]*fm3abscost
                    +
finalStatet4up[1,5]*fm4abscost+finalStatet4up[1,6]*fm5abscost1+finalStatet4up[
1,7]*fm6abscost+finalStatet4up[1,8]*fm7abscost
                    + finalStatet4up[1,9]*flyawaycost1)
abscostriskt5up <-
(finalStatet5up[1,1]*rbrabscost+finalStatet5up[1,2]*fm1abscost1+finalStatet5up
[1,3]*fm2abscost+finalStatet5up[1,4]*fm3abscost
                    +
finalStatet5up[1,5]*fm4abscost+finalStatet5up[1,6]*fm5abscost1+finalStatet5up[
1,7]*fm6abscost+finalStatet5up[1,8]*fm7abscost
                    + finalStatet5up[1,9]*flyawaycost1)
abscostriskt6up <-
(finalStatet6up[1,1]*rbrabscost+finalStatet6up[1,2]*fm1abscost1+finalStatet6up
[1,3]*fm2abscost+finalStatet6up[1,4]*fm3abscost
                    +
finalStatet6up[1,5]*fm4abscost+finalStatet6up[1,6]*fm5abscost1+finalStatet6up[
1,7]*fm6abscost+finalStatet6up[1,8]*fm7abscost
                    + finalStatet6up[1,9]*flyawaycost1)
abscostriskt7up <-
(finalStatet7up[1,1]*rbrabscost+finalStatet7up[1,2]*fm1abscost1+finalStatet7up
[1,3]*fm2abscost+finalStatet7up[1,4]*fm3abscost
                    +
finalStatet7up[1,5]*fm4abscost+finalStatet7up[1,6]*fm5abscost1+finalStatet7up[
1,7]*fm6abscost+finalStatet7up[1,8]*fm7abscost
                    + finalStatet7up[1,9]*flyawaycost1)
abscostriskt8up <-
(finalStatet8up[1,1]*rbrabscost+finalStatet8up[1,2]*fm1abscost1+finalStatet8up
[1,3]*fm2abscost+finalStatet8up[1,4]*fm3abscost
                    +
finalStatet8up[1,5]*fm4abscost+finalStatet8up[1,6]*fm5abscost1+finalStatet8up[
1,7]*fm6abscost+finalStatet8up[1,8]*fm7abscost
                    + finalStatet8up[1,9]*flyawaycost1)
abscostriskt9up <-
(finalStatet9up[1,1]*rbrabscost+finalStatet9up[1,2]*fm1abscost1+finalStatet9up
[1,3]*fm2abscost+finalStatet9up[1,4]*fm3abscost
                    +
finalStatet9up[1,5]*fm4abscost+finalStatet9up[1,6]*fm5abscost1+finalStatet9up[
1,7]*fm6abscost+finalStatet9up[1,8]*fm7abscost
                    + finalStatet9up[1,9]*flyawaycost1)
```

```
abscostriskup <- c(0,
abscostriskt1up,abscostriskt2up,abscostriskt3up,abscostriskt4up,abscostriskt5u
p,abscostriskt6up,abscostriskt7up,
                   abscostriskt8up,abscostriskt9up)

##################################
abscostriskt1new1 <-
(finalStatet1new1[1,1]*rbrabscost+finalStatet1new1[1,2]*fm1abscost2+finalState
t1new1[1,3]*fm2abscost+finalStatet1new1[1,4]*fm3abscost
                      +
finalStatet1new1[1,5]*fm4abscost+finalStatet1new1[1,6]*fm5abscost1+finalStatet
1new1[1,7]*fm6abscost+finalStatet1new1[1,8]*fm7abscost
                      + finalStatet1new1[1,9]*flyawaycost2fm1)
abscostriskt2new1 <-
(finalStatet2new1[1,1]*rbrabscost+finalStatet2new1[1,2]*fm1abscost2+finalState
t2new1[1,3]*fm2abscost+finalStatet2new1[1,4]*fm3abscost
                      +
finalStatet2new1[1,5]*fm4abscost+finalStatet2new1[1,6]*fm5abscost1+finalStatet
2new1[1,7]*fm6abscost+finalStatet2new1[1,8]*fm7abscost
                      + finalStatet2new1[1,9]*flyawaycost2fm1)
abscostriskt3new1 <-
(finalStatet3new1[1,1]*rbrabscost+finalStatet3new1[1,2]*fm1abscost2+finalState
t3new1[1,3]*fm2abscost+finalStatet3new1[1,4]*fm3abscost
                      +
finalStatet3new1[1,5]*fm4abscost+finalStatet3new1[1,6]*fm5abscost1+finalStatet
3new1[1,7]*fm6abscost+finalStatet3new1[1,8]*fm7abscost
                      + finalStatet3new1[1,9]*flyawaycost2fm1)
abscostriskt4new1 <-
(finalStatet4new1[1,1]*rbrabscost+finalStatet4new1[1,2]*fm1abscost2+finalState
t4new1[1,3]*fm2abscost+finalStatet4new1[1,4]*fm3abscost
                      +
finalStatet4new1[1,5]*fm4abscost+finalStatet4new1[1,6]*fm5abscost1+finalStatet
4new1[1,7]*fm6abscost+finalStatet4new1[1,8]*fm7abscost
                      + finalStatet4new1[1,9]*flyawaycost2fm1)
abscostriskt5new1 <-
(finalStatet5new1[1,1]*rbrabscost+finalStatet5new1[1,2]*fm1abscost2+finalState
t5new1[1,3]*fm2abscost+finalStatet5new1[1,4]*fm3abscost
                      +
finalStatet5new1[1,5]*fm4abscost+finalStatet5new1[1,6]*fm5abscost1+finalStatet
5new1[1,7]*fm6abscost+finalStatet5new1[1,8]*fm7abscost
                      + finalStatet5new1[1,9]*flyawaycost2fm1)
abscostriskt6new1 <-
(finalStatet6new1[1,1]*rbrabscost+finalStatet6new1[1,2]*fm1abscost2+finalState
t6new1[1,3]*fm2abscost+finalStatet6new1[1,4]*fm3abscost
                      +
finalStatet6new1[1,5]*fm4abscost+finalStatet6new1[1,6]*fm5abscost1+finalStatet
6new1[1,7]*fm6abscost+finalStatet6new1[1,8]*fm7abscost
                      + finalStatet6new1[1,9]*flyawaycost2fm1)
abscostriskt7new1 <-
(finalStatet7new1[1,1]*rbrabscost+finalStatet7new1[1,2]*fm1abscost2+finalState
t7new1[1,3]*fm2abscost+finalStatet7new1[1,4]*fm3abscost
                      +
finalStatet7new1[1,5]*fm4abscost+finalStatet7new1[1,6]*fm5abscost1+finalStatet
```

```
7new1[1,7]*fm6abscost+finalStatet7new1[1,8]*fm7abscost
                      + finalStatet7new1[1,9]*flyawaycost2fm1)
abscostriskt8new1 <-
(finalStatet8new1[1,1]*rbrabscost+finalStatet8new1[1,2]*fm1abscost2+finalState
t8new1[1,3]*fm2abscost+finalStatet8new1[1,4]*fm3abscost
                      +
finalStatet8new1[1,5]*fm4abscost+finalStatet8new1[1,6]*fm5abscost1+finalStatet
8new1[1,7]*fm6abscost+finalStatet8new1[1,8]*fm7abscost
                      + finalStatet8new1[1,9]*flyawaycost2fm1)
abscostriskt9new1 <-
(finalStatet9new1[1,1]*rbrabscost+finalStatet9new1[1,2]*fm1abscost2+finalState
t9new1[1,3]*fm2abscost+finalStatet9new1[1,4]*fm3abscost
                      +
finalStatet9new1[1,5]*fm4abscost+finalStatet9new1[1,6]*fm5abscost1+finalStatet
9new1[1,7]*fm6abscost+finalStatet9new1[1,8]*fm7abscost
                      + finalStatet9new1[1,9]*flyawaycost2fm1)
abscostrisknew1 <- c(0,
abscostriskt1new1,abscostriskt2new1,abscostriskt3new1,abscostriskt4new1,abscos
triskt5new1,abscostriskt6new1,abscostriskt7new1,
                     abscostriskt8new1,abscostriskt9new1)

##################################
abscostriskt1new5 <-
(finalStatet1new5[1,1]*rbrabscost+finalStatet1new5[1,2]*fm1abscost1+finalState
t1new5[1,3]*fm2abscost+finalStatet1new5[1,4]*fm3abscost
                      +
finalStatet1new5[1,5]*fm4abscost+finalStatet1new5[1,6]*fm5abscost2+finalStatet
1new5[1,7]*fm6abscost+finalStatet1new5[1,8]*fm7abscost
                      + finalStatet1new5[1,9]*flyawaycost2fm5)
abscostriskt2new5 <-
(finalStatet2new5[1,1]*rbrabscost+finalStatet2new5[1,2]*fm1abscost1+finalState
t2new5[1,3]*fm2abscost+finalStatet2new5[1,4]*fm3abscost
                      +
finalStatet2new5[1,5]*fm4abscost+finalStatet2new5[1,6]*fm5abscost2+finalStatet
2new5[1,7]*fm6abscost+finalStatet2new5[1,8]*fm7abscost
                      + finalStatet2new5[1,9]*flyawaycost2fm5)
abscostriskt3new5 <-
(finalStatet3new5[1,1]*rbrabscost+finalStatet3new5[1,2]*fm1abscost1+finalState
t3new5[1,3]*fm2abscost+finalStatet3new5[1,4]*fm3abscost
                      +
finalStatet3new5[1,5]*fm4abscost+finalStatet3new5[1,6]*fm5abscost2+finalStatet
3new5[1,7]*fm6abscost+finalStatet3new5[1,8]*fm7abscost
                      + finalStatet3new5[1,9]*flyawaycost2fm5)
abscostriskt4new5 <-
(finalStatet4new5[1,1]*rbrabscost+finalStatet4new5[1,2]*fm1abscost1+finalState
t4new5[1,3]*fm2abscost+finalStatet4new5[1,4]*fm3abscost
                      +
finalStatet4new5[1,5]*fm4abscost+finalStatet4new5[1,6]*fm5abscost2+finalStatet
4new5[1,7]*fm6abscost+finalStatet4new5[1,8]*fm7abscost
                      + finalStatet4new5[1,9]*flyawaycost2fm5)
abscostriskt5new5 <-
(finalStatet5new5[1,1]*rbrabscost+finalStatet5new5[1,2]*fm1abscost1+finalState
t5new5[1,3]*fm2abscost+finalStatet5new5[1,4]*fm3abscost
```

```
                      +
finalStatet5new5[1,5]*fm4abscost+finalStatet5new5[1,6]*fm5abscost2+finalStatet
5new5[1,7]*fm6abscost+finalStatet5new5[1,8]*fm7abscost
                      + finalStatet5new5[1,9]*flyawaycost2fm5)
abscostriskt6new5 <-
(finalStatet6new5[1,1]*rbrabscost+finalStatet6new5[1,2]*fm1abscost1+finalState
t6new5[1,3]*fm2abscost+finalStatet6new5[1,4]*fm3abscost
                      +
finalStatet6new5[1,5]*fm4abscost+finalStatet6new5[1,6]*fm5abscost2+finalStatet
6new5[1,7]*fm6abscost+finalStatet6new5[1,8]*fm7abscost
                      + finalStatet6new5[1,9]*flyawaycost2fm5)
abscostriskt7new5 <-
(finalStatet7new5[1,1]*rbrabscost+finalStatet7new5[1,2]*fm1abscost1+finalState
t7new5[1,3]*fm2abscost+finalStatet7new5[1,4]*fm3abscost
                      +
finalStatet7new5[1,5]*fm4abscost+finalStatet7new5[1,6]*fm5abscost2+finalStatet
7new5[1,7]*fm6abscost+finalStatet7new5[1,8]*fm7abscost
                      + finalStatet7new5[1,9]*flyawaycost2fm5)
abscostriskt8new5 <-
(finalStatet8new5[1,1]*rbrabscost+finalStatet8new5[1,2]*fm1abscost1+finalState
t8new5[1,3]*fm2abscost+finalStatet8new5[1,4]*fm3abscost
                      +
finalStatet8new5[1,5]*fm4abscost+finalStatet8new5[1,6]*fm5abscost2+finalStatet
8new5[1,7]*fm6abscost+finalStatet8new5[1,8]*fm7abscost
                      + finalStatet8new5[1,9]*flyawaycost2fm5)
abscostriskt9new5 <-
(finalStatet9new5[1,1]*rbrabscost+finalStatet9new5[1,2]*fm1abscost1+finalState
t9new5[1,3]*fm2abscost+finalStatet9new5[1,4]*fm3abscost
                      +
finalStatet9new5[1,5]*fm4abscost+finalStatet9new5[1,6]*fm5abscost2+finalStatet
9new5[1,7]*fm6abscost+finalStatet9new5[1,8]*fm7abscost
                      + finalStatet9new5[1,9]*flyawaycost2fm5)
abscostrisknew5 <- c(0,
abscostriskt1new5,abscostriskt2new5,abscostriskt3new5,abscostriskt4new5,abscos
triskt5new5,abscostriskt6new5,abscostriskt7new5,
                     abscostriskt8new5,abscostriskt9new5)

#######################
# Purpose: Calculate the average cost delta between the estimate and the new
design alternative
# inputs: abscostriskest[1-9], abscostrisknew1[1-9], abscostrisknew5[1-9]
# outputs: avgcostdeltafm1, avgcostdeltafm5
# Author: Bryan Bentz
# Notes: I printed this onto the graphs of abscostrisk for new fm1 and new
fm5.
#        By evaluating the cost risk section to minimize the costriskdelta's I
found the costs
#        that each subsystem must be based on the Delta1 and Delta5.
#        This added in the generation of Figures 40-44.
#######################

costdeltafm1 <- c((abscostriskest[1]-abscostrisknew1[1]),
                  (abscostriskest[2]-abscostrisknew1[2]),
                  (abscostriskest[3]-abscostrisknew1[3]),
                  (abscostriskest[4]-abscostrisknew1[4]),
                  (abscostriskest[5]-abscostrisknew1[5]),
                  (abscostriskest[6]-abscostrisknew1[6]),
                  (abscostriskest[7]-abscostrisknew1[7]),
                  (abscostriskest[8]-abscostrisknew1[8]),
                  (abscostriskest[9]-abscostrisknew1[9]))

averagecostdeltafm1 <- sum(costdeltafm1)/length(costdeltafm1)

costdeltafm5 <- c((abscostriskest[1]-abscostrisknew5[1]),
                  (abscostriskest[2]-abscostrisknew5[2]),
                  (abscostriskest[3]-abscostrisknew5[3]),
                  (abscostriskest[4]-abscostrisknew5[4]),
                  (abscostriskest[5]-abscostrisknew5[5]),
                  (abscostriskest[6]-abscostrisknew5[6]),
                  (abscostriskest[7]-abscostrisknew5[7]),
                  (abscostriskest[8]-abscostrisknew5[8]),
                  (abscostriskest[9]-abscostrisknew5[9]))

averagecostdeltafm5 <- sum(costdeltafm5)/length(costdeltafm5)

x=c(0,1,2,3,4,5,6,7,8,9)
par(col='black')
plot(x=x, y=abscostriskest,ylim=c(40000,160000),main="Baseline Absolute Cost
Risk",
     xlab='Sortie',ylab='Cost at Risk ($)',xlim=c(0,9),axes=F, pch = 19)
xticks <- seq(0, 9, 1)
yticks <- seq(40000,160000,10000)
axis(1, at = xticks, labels = xticks, las=1, tck=-0.01)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(x=x, y=abscostrisklow,ylim=c(40000,160000),ann=F,xlim=c(0,9),axes=F, pch
= 24)
par(new=TRUE)
plot(x=x, y=abscostriskup,ylim=c(40000,160000),ann=F,xlim=c(0,9),axes=F, pch =
25)
```

```
par(new=FALSE)

x=c(0,1,2,3,4,5,6,7,8,9)
par(col='black')
plot(x=x, y=abscostriskest,ylim=c(40000,160000),main="Absolute Cost Risk with
Non-Repairable Propulsion",
     xlab='Sortie',ylab='Cost at Risk ($)',xlim=c(0,9),axes=F, pch = 19)
xticks <- seq(0, 9, 1)
yticks <- seq(40000,160000,10000)
axis(1, at = xticks, labels = xticks, las=1, tck=-0.01)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(x=x, y=abscostrisklow,ylim=c(40000,160000),ann=F,xlim=c(0,9),axes=F, pch
= 24)
```

```
par(new=TRUE)
plot(x=x, y=abscostriskup,ylim=c(40000,160000),ann=F,xlim=c(0,9),axes=F, pch =
25)
par(new=TRUE)
plot(x=x,
y=abscostrisknonrep5,ylim=c(40000,160000),ann=F,xlim=c(0,9),axes=F,col =
'blue', pch = 15)
```

```
x=c(0,1,2,3,4,5,6,7,8,9)
par(col='black')
plot(x=x, y=abscostriskest,ylim=c(40000,160000),main="Absolute Cost Risk with
Non-Repairable Structure",
     xlab='Sortie',ylab='Cost at Risk ($)',xlim=c(0,9),axes=F, pch = 19)
xticks <- seq(0, 9, 1)
yticks <- seq(40000,160000,10000)
axis(1, at = xticks, labels = xticks, las=1, tck=-0.01)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(x=x, y=abscostrisklow,ylim=c(40000,160000),ann=F,xlim=c(0,9),axes=F, pch
= 24)
par(new=TRUE)
plot(x=x, y=abscostriskup,ylim=c(40000,160000),ann=F,xlim=c(0,9),axes=F, pch =
25)
par(new=TRUE)
plot(x=x,
y=abscostrisknonrep7,ylim=c(40000,160000),ann=F,xlim=c(0,9),axes=F,col =
'green', pch = 18)
```

```
x=c(0,1,2,3,4,5,6,7,8,9)
par(col='black')
plot(x=x, y=abscostriskest,ylim=c(40000,120000),main="Absolute Cost Risk w/
Altered Electronics h(t) and $$",
     xlab='Sortie',ylab='Cost at Risk ($)',xlim=c(0,9),axes=F, pch = 19)
xticks <- seq(0, 9, 1)
yticks <- seq(40000,120000,10000)
axis(1, at = xticks, labels = xticks, las=1, tck=-0.01)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(x=x, y=abscostrisklow,ylim=c(40000,120000),ann=F,xlim=c(0,9),axes=F, pch
= 24)
par(new=TRUE)
plot(x=x, y=abscostriskup,ylim=c(40000,120000),ann=F,xlim=c(0,9),axes=F, pch =
25)
par(new=TRUE)
plot(x=x, y=abscostrisknew1,ylim=c(40000,120000),ann=F,xlim=c(0,9),axes=F, col
= 'red', pch = 19)
par(new=TRUE)
mylabel <- bquote(italic(avgdeltacostrisk) == .(format(averagecostdeltafm1,
digits = 2)))
legend('bottomright', legend = mylabel, bty='n')
```

```
par(new=FALSE)

x=c(0,1,2,3,4,5,6,7,8,9)
par(col='black')
plot(x=x, y=abscostriskest,ylim=c(40000,120000),main="Absolute Cost Risk w/
Altered Propulsion h(t) and $$",
     xlab='Sortie',ylab='Cost at Risk ($)',xlim=c(0,9),axes=F, pch = 19)
xticks <- seq(0, 9, 1)
yticks <- seq(40000,120000,10000)
axis(1, at = xticks, labels = xticks, las=1, tck=-0.01)#was -0.01
axis(2,las=1,at=yticks,labels=yticks)
par(new=TRUE)
plot(x=x, y=abscostrisklow,ylim=c(40000,120000),ann=F,xlim=c(0,9),axes=F, pch
= 24)
par(new=TRUE)
plot(x=x, y=abscostriskup,ylim=c(40000,120000),ann=F,xlim=c(0,9),axes=F, pch =
25)
par(new=TRUE)
plot(x=x, y=abscostrisknew5,ylim=c(40000,120000),ann=F,xlim=c(0,9),axes=F, col
= 'blue', pch = 19)
par(new=TRUE)
mylabel <- bquote(italic(avgdeltacostrisk) == .(format(averagecostdeltafm5,
digits = 2)))
legend('bottomright', legend = mylabel, bty='n')
```

```
par(new=FALSE)

#######################
# Purpose: Visual Representation of Markov Chains to come
# inputs: N/A
# outputs: Example Markov Chain Model with Identified Subsystems, Figure 12
# Author: Bryan Bentz
# notes:
#######################

Mat2 <- matrix(NA, nrow = 9, ncol = 9)

AA <- as.data.frame(Mat2)
AA[[1,2]] <- 'F[1:0]'
AA[[1,3]] <- 'F[2:0]'
AA[[1,4]] <- 'F[3:0]'
AA[[1,5]] <- 'F[4:0]'
AA[[1,6]] <- 'F[5:0]'
AA[[1,7]] <- 'F[6:0]'
AA[[1,8]] <- 'F[7:0]'
AA[[2,1]] <- 'F[0:1]'
AA[[3,1]] <- 'F[0:2]'
AA[[4,1]] <- 'F[0:3]'
AA[[5,1]] <- 'F[0:4]'
AA[[6,1]] <- 'F[0:5]'
```

```
AA[[7,1]] <- 'F[0:6]'
AA[[8,1]] <- 'F[0:7]'
AA[[9,2]] <- 'F[1:8]'
AA[[9,3]] <- 'F[2:8]'
AA[[9,4]] <- 'F[3:8]'
AA[[9,5]] <- 'F[4:8]'
AA[[9,6]] <- 'F[5:8]'
AA[[9,7]] <- 'F[6:8]'
AA[[9,8]] <- 'F[7:8]'
AA[[9,9]] <- 'F[8:8]'
AA[[1,1]] <- 'F[0:0]'

AA <- as.data.frame(Mat2)
AA[[1,2]] <- ''
AA[[1,3]] <- 'Repair'
AA[[1,4]] <- 'Repair'
AA[[1,5]] <- 'Repair'
AA[[1,6]] <- 'Repair'
AA[[1,7]] <- 'Repair'
AA[[1,8]] <- 'Repair'
AA[[2,1]] <- ''
AA[[3,1]] <- 'Failure'
AA[[4,1]] <- 'Failure'
AA[[5,1]] <- 'Failure'
AA[[6,1]] <- 'Failure'
AA[[7,1]] <- 'Failure'
AA[[8,1]] <- 'Failure'
AA[[9,2]] <- 'Destruction'
AA[[9,3]] <- 'Destruction'
AA[[9,4]] <- 'Destruction'
AA[[9,5]] <- 'Destruction'
AA[[9,6]] <- 'Destruction'
AA[[9,7]] <- 'Destruction'
AA[[9,8]] <- ''
AA[[9,9]] <- 'Destroyed'
AA[[1,1]] <- 'Regen'

names <- c('Operational','Electronics','Fuel
Mgmt','Launcher','Operator','Propulsion','Recovery',
           'Structure','Failed')

par(family='serif', mar = c(0,0,0,0))

diagram::plotmat(A = AA, pos = 9, curve = .65,
                 name = names, lwd =1, arr.len = 0.3,
                 arr.width = 0.15, my = 0, box.size = 0.05,
                 arr.type = 'triangle', dtext = -1.0,
                 relsize=.9,box.cex=0.7,cex=1)
```

Appendix B: MTBF Results if Simplification is Acceptable

This research discusses the weaknesses of mean time between failures (MTBF) as a sole reliability metric as the use of the mean assumes a constant hazard rate which masks the manner in which failures actually occur over time. This hides failures due to infant mortality and failure due to wear out. The use of reliability metrics are often inputs for other decision makers – especially those responsible for tasks such as cost estimation. In these situations, true estimates of MTBF based on failure data are more useful than estimates from the manufacturer or other systems. However, based on analysis presented in Table 5 the failure-time data generated through the assumption of a constant hazard rate, and its representation in terms of MTBF, could be as much as 47% off from a metric that is allowed to vary with time. If this disadvantage is deemed acceptable in order to achieve a data-based estimate to include for cost estimation purposes, the cumulative incidence functions can be fit to a one parameter exponential distribution using fitdistr in R. In this case the parameter, λ, is defined as the inverse of the MTBF. Applying this analysis to the baseline system's failure-time data yields:

Table 6: Estimated MTBF of Seven Subsystems

Subsystem	Mean Sorties Between Failure
Electronics	62.3 Sorties
Fuel Management	88.05 Sorties
Launcher	111.8 Sorties
Operator	616.1 Sorties
Propulsion	244.3 Sorties
Recovery	775.5 Sorties
Structural	1594.95 Sorties

These estimates must be used cautiously as they predict MTBFs of sixty or more sorties, yet the failure-time data for this research was only complete enough to use out to nine sorties. It is likely that only increases the error induced by accepting the constant hazard rate assumption. Furthermore, if these disadavantages are still acceptable, these MTBFs can be combined as the exponential distributions of the subsystems can be added to determine the MTBF of the system as a whole. It was determined that the system-level MTBF was 22.7 Sorties. This statistic accounts for the nature of the competing failure modes as well as the right censoring scheme to determine the system's MTBF for failures of these seven subsystems.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
23-03-2017	Master's Thesis	August 2015 – March 2017

4. TITLE AND SUBTITLE

Reliability and Cost Impacts for Attritable Systems

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Bentz, Bryan R., 1st Lieutenant, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 640
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT-ENV-MS-17-M-172

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Aerospace Vehicles Division, Air Force Research Laboratories
2145 5th Street, Bldg. 24C, WPAFB, OH
jason.w.sutherlin@us.af.mil; 937-656-8786

10. SPONSOR/MONITOR'S ACRONYM(S)

AFRL/RQVI

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

This material is declared a work of the U. S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

Attritable systems trade system attributes like reliability and reparability to achieve lower acquisition cost and decrease cost risk. Ultimately, it is hoped that by trading these attributes the amount of systems able to be acquired will be increased. However, the effect of trading these attributes on system-level reliability and cost risk is difficult to express complicated reparable systems like an air vehicle. Failure-time and cost data from a baseline limited-life air vehicle is analyzed for this reliability and reparability trade study. The appropriateness of various reliability and cost estimation techniques are examined for these data. This research employs the cumulative incidence function as an input to discrete time non-homogeneous Markov chain models to overcome the hurdles of representing the failure-time data of a reparable system with competing failure modes that vary with time. This research quantifies the probability of system survival to a given sortie, $S(n)$, average unit flyaway cost (AUFC), and cost risk metrics to

convey the value of reliability and reparability trades. Investigation of the benefit of trading system reparability shows a marked increase in cost risk. Yet, trades in subsystem reliability calculate the required decrease in subsystem cost required to make such a trade advantageous. This research results in a trade-space analysis tool that can be used to guide the development of future attritable air vehicles.

15. SUBJECT TERMS

Reliability, Reparability

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Dr. John Colombi (ENV)
					19b. TELEPHONE NUMBER *(Include area code)*
U	U	U	UU	186	(937) 255-3636 x3347 john.colombi@afit.edu