Design of Autonomous Systems for Survivability through Conceptual Object-Based Risk Analysis

By

Ada Rhodes Short
A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science (Mechanical Engineering).

Golden, Colorado

Date ________________________

Signed: __________________________

Ada Rhodes Short

Golden, Colorado

Date ________________________

Signed: __________________________

Dr. Douglas L Van Bossuyt
Thesis Advisor

Golden, Colorado

Date ________________________

Signed: __________________________

Dr. Greg Jackson
Professor and Head of the Department of Mechanical Engineering
ABSTRACT

As human space exploration reaches the outer planets and other isolated environments through the use of autonomous and semi-autonomous systems, it becomes increasingly important that the autonomous systems be designed for survivability under unknown and hazardous conditions. Existing methods for autonomous system design for survivability are limited in their ability to adapt to unknown environments and do not use a combined control-design iteration in the early conceptual stage of design for increased survivability during system operations. To address the problem of control-design integration for survivability of an autonomous system, a functional model of the autonomous system of interest and the operating environment are constructed and analyzed using a system-of-systems approach. The Conceptual Object-Based Risk Analysis (COBRA) design methodology is a proposed four-phase iterating method that begins with early phase functional models of an autonomous system design that is analyzed to determine how potential failure may propagate through the autonomous system. Next, methods for integrated navigational control decisions with the physical design are tuned for use with the conceptual design. Then, an analysis of the proposed autonomous system’s fitness for a planned mission is evaluated. Finally, a physical and control design is proposed that is iterated upon starting from the first phase. The research presented in this thesis focuses on two case studies. The first case study focus is the control and conceptual design of an autonomous planetary rover and demonstrates the effectiveness of the COBRA method for improving an autonomous system’s ability to survive in a simulated Mars-like environment. The second case study adapts the COBRA method for application on a human-centered space mission modeled as a system-of-systems. The COBRA method has potential applications in space exploration, sea floor exploration and surveying, self-driving cars, space mission planning and analysis, and industrial automation.
# TABLE OF CONTENTS

Abstract ....................................................................................................................................... iii  
List of Figures .............................................................................................................................. viii  
List of Tables .............................................................................................................................. x  
List of Symbols .............................................................................................................................. xi  
Abbreviations and Acronyms ...................................................................................................... xiii  
Acknowledgements ....................................................................................................................... xv  
Chapter 1 Introduction .................................................................................................................... 1  
  1.1 Specific Contributions ............................................................................................... 3  
Chapter 2 Background and Previous Work .................................................................................... 6  
  2.1 Background ........................................................................................................... 6  
    2.1.1 Systems and Functional Modeling ................................................................. 6  
    2.1.2 Failure Analysis Methods ............................................................................... 7  
    2.1.3 Prognostics and Health Management ............................................................ 8  
    2.1.4 Risk Attitude ..................................................................................................... 8  
    2.1.5 Space Mission Risk Assessment .................................................................... 9  
    2.1.6 Autonomous Mobility in Robotics ............................................................... 9  
    2.1.7 Route Planning ............................................................................................... 10  
  2.2 Previous Work ......................................................................................................... 10  
    2.2.1 Failure Flow Decision Functions .................................................................... 11  
    2.2.2 Risk Informed Navigational Controls ........................................................... 12  
Chapter 3 COBRA Methods and Their Application ..................................................................... 20  
  Phase 1) Modeling Analysis of System ................................................................. 21  
  Phase 2) Generate PHM-Informed Controls ........................................................... 21
Phase 3) Analyze System Mission Effectiveness .......................................................... 22
Phase 4) Integrate Elements into a Final Physical Design ........................................... 22

Chapter 4 Failure Flow Decision Functions for Iterative Design ........................................ 25

Abstract .......................................................................................................................... 25

4.1 Introduction ............................................................................................................... 25
  4.1.1 Specific Contributions ....................................................................................... 27

4.2 Background ............................................................................................................... 28
  4.2.1 System and Functional Modeling .................................................................. 28
  4.2.2 Failure Analysis Methods ............................................................................. 29
  4.2.3 Hazard Rate Estimation and Success Assessment ........................................... 30
  4.2.4 Failure Flow Decision Functions .................................................................. 31

4.3 Methodology ............................................................................................................ 32
  Phase 1: Generate Functional Model ........................................................................ 32
  Phase 2: Design Iteration ......................................................................................... 36
  Phase 3: Interpret and Report Results ...................................................................... 45

4.4 Case Study ............................................................................................................... 46
  Step 1: Develop Functional Model ........................................................................ 46

4.5 Results ..................................................................................................................... 51

4.6 Discussion ............................................................................................................... 53

4.7 Conclusion and Future Work ................................................................................ 54

Acknowledgements ........................................................................................................ 56

Chapter 5 Active Mission Success Estimation Through PHM- Informed Probabilistic Modelling for a Crewed Martian Mission ................................................................. 57

Abstract ......................................................................................................................... 57

5.1 Introduction ............................................................................................................... 58
  5.1.1 Specific Contributions .................................................................................... 59
5.2 Background ........................................................................................................................................ 60
- 5.2.1 Functional Modelling .................................................................................................................. 60
- 5.2.2 Space Mission Risk Assessment ................................................................................................ 61
- 5.2.3 Prognostics and Health Management ......................................................................................... 62

5.3 Methodology ....................................................................................................................................... 62
- Pre Steps ............................................................................................................................................... 63
- Phase 1: Modelling ................................................................................................................................. 64
- Phase 2: AMSE Analysis ......................................................................................................................... 71
- Phase 3: Interpretation of Results .......................................................................................................... 78

5.4 Case Study ......................................................................................................................................... 79
- 5.4.1 Crew Composition ....................................................................................................................... 79
- 5.4.2 Human Requirements to Live in Space ....................................................................................... 80
- 5.4.3 Human Exploration of Mars and Site Selection ........................................................................ 80
- 5.4.4 Nutrition Requirements ............................................................................................................. 81
- 5.4.5 Included Model Systems and Resources .................................................................................. 82
- 5.4.6 Mission Plan ................................................................................................................................. 84
- 5.4.7 AMSE Cases ................................................................................................................................. 84

5.5 Results and Discussion ..................................................................................................................... 87
- 5.5.1 Inaccurate Mission Calculations ................................................................................................. 87
- 5.5.2 Inability to Farm .......................................................................................................................... 87
- 5.5.3 Broken Arm ............................................................................................................................... 93
- 5.5.4 Discussion of Results .................................................................................................................. 95

5.6 Conclusion and Future Work ........................................................................................................... 96
- 5.6.1 Future Work ............................................................................................................................... 96

Acknowledgements ............................................................................................................................... 97
<table>
<thead>
<tr>
<th>Chapter</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 6</td>
<td>Discussion and Broader Impacts</td>
<td>98</td>
</tr>
<tr>
<td>6.1</td>
<td>Failure Flow Decision Functions (FFDF)</td>
<td>98</td>
</tr>
<tr>
<td>6.2</td>
<td>COBRA-Based Navigational Controls</td>
<td>100</td>
</tr>
<tr>
<td>6.3</td>
<td>Active Mission Success Estimation (AMSE)</td>
<td>101</td>
</tr>
<tr>
<td>6.4</td>
<td>Broader Impacts</td>
<td>102</td>
</tr>
<tr>
<td>Chapter 7</td>
<td>Future Work</td>
<td>104</td>
</tr>
<tr>
<td>Chapter 8</td>
<td>Conclusion</td>
<td>106</td>
</tr>
<tr>
<td>References</td>
<td></td>
<td>108</td>
</tr>
<tr>
<td>APPENDIX A.1</td>
<td></td>
<td>116</td>
</tr>
<tr>
<td>APPENDIX A.2</td>
<td></td>
<td>117</td>
</tr>
<tr>
<td>APPENDIX A.3</td>
<td></td>
<td>118</td>
</tr>
<tr>
<td>APPENDIX B.1</td>
<td></td>
<td>119</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1.1: Flow for COBRA based Design................................................................. 1

Figure 2.1: Process flow for RAIR method ............................................................... 13

Figure 2.2: Plots of rover paths on four maps, for risk tolerance $\rho$. YELLOW: $\rho=5$, BLUE: $\rho=10$, RED: $\rho=20$, GREEN: $\rho=50$................................................................. 15

Figure 2.3: Process flow of the PIDAA method......................................................... 16

Figure 2.4: Red: a control rover taking the most topographically efficient path, Blue: PIDAA controlled rover driving away from danger......................................................... 17

Figure 2.5: Process flow of GLPFDOE method .......................................................... 18

Figure 3.1: The Conceptual Object Based Risk Analysis Method ............................... 20

Figure 3.2: (LEFT) The 2nd, 3rd, and 4th generation ROBRT designs, (RIGHT) The 1st and 2nd generation PETR designs................................................................. 23

Figure 4.1: Process Flow of FFDF-ID ........................................................................ 33

Figure 4.2: An example simplified functional model of a Mars rover. Rectangular boxes represent functions, thin lines represent energy flows, thick lines represent material flows, dotted lines represent signals or information........................................ 34

Figure 4.3: Process for solving functional model failure flow path problem ............... 38

Figure 4.4: Process for Calculating total failure probability for an initiating event ........ 42

Figure 5.1: Phase 1, Modelling, Process .................................................................... 65

Figure 5.2: Functional Model of an SEV ................................................................. 66

Figure 5.3: System-of-systems Functional Model .................................................... 66

Figure 5.4: Organization Structure of Tasks ............................................................ 70

Figure 5.5: Phase 2, Analysis, Process Flow ............................................................ 71

Figure 5.6: Sols 0 through 615 of the General Mission Plan ........................................ 85

Figure 5.7: Individual Survival Rate for Inaccurate Caloric Needs ................................ 89

Figure 5.8: Individual Survival Rate for Inaccurate Caloric Needs............................. 90
Figure 5.9: Individual Survival Probabilities If Unable to Farm .................................................. 91

Figure 5.10: Individual Survival Probabilities If Unable to Farm and Food is Rationed .......... 92

Figure 5.11: Individual Survival Probabilities If Unable to Farm and Food is Rationed and
Physical Activity Reduced ......................................................................................... 93

Figure 5.12: Individual Survival Probabilities If Work Allocation is Change to Allow For
Arm to Heal ............................................................................................................... 94
# LIST OF TABLES

Table 2.1: Results of GLPFDOE iteration on a simulated Mars rover ........................................ 19

Table 4.1: Functions Used ....................................................................................................... 47

Table 4.2: Results of FFDF-ID Analysis after 1 round of iteration shown in Probability of Survival on Demand (PSD) ................................................................. 52

Table 4.3: Results of FFDF-ID Analysis after 2 rounds of iteration shown in Probability of Survival on Demand (PSD) ................................................................. 53

Table 5.1: Typical Mission Tasks and Associated Costs ....................................................... 69

Table 5.2: Hazard Rates for Failure Type [Failure/hour] ....................................................... 74

Table 5.3: Crew EVA schedule over a Nine-Sol period ......................................................... 84

Table 5.4: Revised EVA Nine Sol Schedule ......................................................................... 94
LIST OF SYMBOLS

\( \Omega \) – Objective Function Value

\( \theta \) – Heading of navigational target in the rover frame

\( \theta_p \) – Heading of a point of interest in the rover frame

\( \rho \) – Risk tolerance factor

\( \Delta L \) – Distance to target from rover

\( \Delta L_p \) – Distance to target from rover

\( \eta \) – Environmental hazard profile

\( PMC \) – Probable Mission Completion score

\( OY \) – Objective yield

\( w \) – Waypoints reached

\( \lambda \) – Hazard rate

\( t \) – Time

\( S \) – Probability of survival

\( F \) – Probability of failure

\( F^{\text{CritFail}} \) - Probability of critical system failure

\( f \) – A function in a functional model, \( M \)

\( \varphi \) – A flow in a function model, \( M \)

\( M \) – A functional model

\( ICF \) – Independent Critical Function

\( kNCF \) – k of N Critical Function
$p_{i,c}$ – A path through a functional model

$P$ – Set of all paths through a functional model

$p_{f,i,c}$ – A unique path through a functional model beginning at function $f$

$UP_f$ – Set of all unique paths starting from function $f$

$P^{\text{pass}}_{f,\varphi}$ – Probability of function, $f$, passing failure along path, $\varphi$

$P^{\text{accept}}_{f,\varphi}$ – Probability that function, $f$, will accept failure passed along path, $\varphi$

$I_E$ – An initiating failure event

$F_u$ – Probability of critical system failure from an undirected failure entering the system

$\tau$ – Instantaneous moment in time

$BMI$ – Body Mass Index

$a$ – A critical system of interest in a mission

$\Lambda_a$ – Total mission hazard rate presented to system $a$

$\varepsilon$ – An individual task in a task plan, $E$

$E$ – Task plan as a set of all tasks, $\varepsilon$

$h$ – Class of hazards faced by a system of interest

$H_{\varepsilon}$ – Set of all hazards present in the completion of task $\varepsilon$

$p_{a_h}$ – Set of all parameters used to calculate hazard rates in a PHM based failure distribution

$T$ – Total planned mission length

$\Delta \tau_{\varepsilon}$ – Time elapsed during the completions of task, $\varepsilon$

$S_a(t)$ – Probability of survival of system, $a$, at time, $t$

$p^{\text{success}}(t)$ – Probability of total mission success at time, $t$
ABBREVIATIONS AND ACRONYMS

AI – Artificial Intelligence
AMSE – Active Mission Success Estimation
AMSEAD – Active Mission Success Estimation for Autonomous Decisions
BMI – Body Mass Index
CDF – Cumulative Distribution Function
CDI – Control Design Integration
COBRA – Conceptual Object-Based Risk Analysis
DRV – Daily Recommended Value
EFFBD – Enhanced Function Flow Block Diagrams
EMU – Extravehicular Mobility Units
EVA – Extra Vehicular Activities
FBED – Functional Basis for Engineering Design
FFD – Functional Flow Diagrams
FFDF – Failure Flow Decision Functions
FFIP – Function Failure Identification and Propagation
FMEA – Failure Modes and Effects Analysis
GLPFDOE – Global to Local Path Finding Design and Operation Exploration
GUI – Graphical User Interface
ICF – Independent Critical Functions
ISRU – In-Situ Resource Utilization
IVA – Intra-Vehicular Activities
LED – Light Emitting Diode
MED – Mars Exploration Rover
MOC – Mars Orbiter Camera
NASA – National Aeronautics and Space Administration
PETR – PHM Enabled Test Rover
PHM – Prognostics and Health Management
PIDAA – PHM Informed Damage Aversion Algorithms
PRA – Probabilistic Risk Assessment
PSD – Probability of Survival on Demand
RAIR – Risk Attitude Informed Route-Planning
ROBRT – Robotic Optimized Bathyscaphe Reliability Testbed
ROV – Remote Operated Vehicle
SEV – Surface Exploration Vehicle
SPEARS – Simulated Physics and Environment for Autonomous Risk Studies
UAV – Uncrewed Aerial Vehicle
ACKNOWLEDGEMENTS

My work is the result of a long line of friends, mentors, teachers, and professors who continuously pushed me farther and helped me reach higher than I ever thought I could. Thank you to my advisor Dr. Douglas Van Bossuyt, who gave me the opportunity to ask questions and the privilege of a graduate education. I also would like to thank all of the scholars who came before me and made my work possible, as well as the undergraduate students that worked tirelessly in my lab to support my work and make my research possible, thank you Zachary Mimlitz, Alexis Humann, David Hodge, Robin Coleman, Keenan Murphy, Kevin Nelson, and Sami Al-Saadawi. I owe a great debt to all the people that helped to keep me on track, thank you Liana and Sherman Sanders, Len Joeris, Helen Rajala, Todd Jones, Elaine Podell, Jeff Blume, Joe and Suzette Greany, and Dr. Carolyn Skurla, I owe you the world. Thank you to my friends for always supporting and inspiring me, and thank you to my boyfriend for being so patient, loving, and understanding over the past 6 years. Thank you to my parents for giving me life. Finally, I again wish to deeply and sincerely thank my advisor for his constant support, patience, and wisdom.
CHAPTER 1
INTRODUCTION

Conceptual Object-Based Risk Analysis (COBRA) is a family of methods that aims to enable the design of autonomous systems for improved survivability in high risk and hazardous environments. COBRA is based heavily in functional modeling and Prognostics and Health Management (PHM) techniques, and can be categorized into four major phases; 1) early phase functional modeling of an autonomous system design for analysis in order to determine how potential failure may propagate through the autonomous system, 2) integration of functional models into control systems for PHM informed decision making and navigation, 3) analysis of the proposed autonomous system’s fitness for a planned mission, and 4) a physical and control design is proposed based on insight gleamed from the previous phases, Phase 4 can then be iterated on by closing the loop and returning to Phase 1 with a functional model of the new system design. Figure 1.1 provides a flow chart of the COBRA process with arrows representing the flow of information between Phases.

Figure 1.1: Flow for COBRA based Design

Note that each of the first three phases feeds into the next phase of analysis as well as connecting directly to the fourth phase. This is representative of how each individual phase of the COBRA method can provide insight into system design, but each subsequent phase is dependent on the phases that came before it. COBRA is an iterative design process that can be iterated upon many times to achieve desired results.
Chapter 4 of this thesis presents the Modeling System Failure Decisions (Phase 1) portion of COBRA through the Failure Flow Decision Function for Iterative Design (FFDF-ID) [1] methodology in the form of a journal article in preparation for publication, which builds upon previous work performed towards the development of Failure Flow Decision Functions (FFDF) [2], discussed in Chapter 2.

The PHM-Informed Controls portion (Phase 2) of COBRA was explored in a variety of previously published conference papers [3–6], several of which are summarized in the Previous Work section of Chapter 2, Section 2.2.2 of this thesis. These methods explored the effects of varying risk attitudes and autonomous behaviors towards the effectiveness of survival and navigational decisions. This was done by quantifying a Risk Attitude (RA) [7] and building it into an objective function used for making decisions. The Global to Local Path Finding Design and Operation Exploration (GLPFDOE) method [6] embodies the path between Phase 2 and Phase 4 of COBRA.

Complexity of COBRA-based functional models was explored through the Active Mission Success Estimation (AMSE) methodology which is presented in Chapter 5 as a journal article in preparation for publication. AMSE takes a more complex approach to the problem by viewing a system-of-systems that is made up of a variety of components such as a space mission consisting of rovers, launch vehicles, communication systems, and an environmental model. Additionally, the AMSE papers looks into how broadly COBRA could be applied by investigating a case study that views humans as the primary system of interest in a system-of-systems containing a variety of electrical and mechanical systems. Including humans as a central focus of the analysis is important to understanding how systems can be designed that consider their effects on human operators, collaborators, or bystanders in the general area of operation, enabling for design and operation for improved system safety. AMSE is representative of a Mission Success Estimation portion (Phase 3) COBRA method for mission success estimation. AMSE is informed by Phase 1 and Phase 2 modelling and analysis of multiple systems that are then constructed into a mission framework. Currently, no studies have been performed on the adaption of AMSE for application to COBRA Phase 4, but future work will attempt to address this problem.
This thesis first reviews general background information and previous work performed necessary to understanding the work presented. It then provides a brief overview of the COBRA method at its current stage in development. Next the thesis presents two journal articles in preparation for publication on FFDF-ID and AMSE. Finally, the paper concludes with discussion of the work presented and future work to be performed.

1.1 Specific Contributions

The work presented in this thesis contains unique contributions to functional modeling, failure analysis, engineering decision making, and design of autonomous systems.

Functional modeling consists of the modeling of the functionality of a system. This is generally performed from a Functional Basis for Engineering Design (FBED) perspective, discussed further in the Chapter 2 of this thesis. This work contributes to the existing FBED lexicon through the inclusion of a direct failure function. This function is representative of subsystems or components with the ability to direct an eminent failure to a desire subsystem. Including this function allows for the analysis of systems that are able to take failure mitigating actions on their own, for example in an autonomous mobile robot a direct failure function may be representative of a circuit for directing an unexpected electrical surge to a particular subsystem in order to protect subsystems more critical to continued operation. Inclusion of this function in a model allows for autonomous decisions to be analyzed when facing failure, by representing available options. Previously, no method existed to explicitly model directed failure flow functions, especially with respect to modelling the function during both nominal and failure conditions.

A second contribution of this work is the creation of nested functional models to represent environments of operation. Existing functional modeling techniques are limited in their ability to consider the effects of an environment on the system’s operation. In the AMSE method presented in Chapter 5 structures the functional models to allow the analysis of environmental effects on the operation of a system-of-systems. In addition to the greater environment of operation, protective layers of micro-environments can be modeled in this way. For example, an astronaut, in a space suit, in a vehicle, in space can be directly considered. In this example every time a flow passes through an environmental layer, it is acted upon by this layer. This allows for the analysis of factors like the reduction in magnitude of radiation present in the environment, insulation of heat, or
protection from physical impact. Introduction of the nested system-of-systems modelling technique allows for the development and analysis of more complex and realistic functional models. Previously, nested functional models were used primarily to reduce complexity presented graphically to a system designer while allowing a system designer to zoom in on hidden details when desired.

Contributions to failure analysis comes from an improved method for analyzing failure propagation and the associated probability of propagation through a functional model of a system. Calculating failure propagation through a system is dependent on two primary steps, discovery of paths, and calculation of probability. Discovery of failure paths is a very difficult problem as thousands of paths can exist for even a comparatively small and simple system. In order to find all of the paths, a biased maze solving algorithm was adapted in order to be able to handle the logical constraints of failure flow through a functional model. A more detailed description of this approach is discussed in Chapter 5 of this thesis. The discovered paths can then be used to calculate the probability of failure propagation from any function in the model. Rapid calculation of failure propagation probabilities is a critical step to enabling the design and operation of autonomous systems in hazardous environments.

The AMSE method presented in Chapter 5 presents a novel method for active mission success estimation. This allows for the analysis of mission success over the course of the mission and is performed rapidly enough that it can be used to inform mission decisions for rapidly developing situations. The AMSE method provides two unique contributions. The first is that AMSE allows a mission to be view as a developing probability of success over time. This allows for initial mission success to be maximized and informs the individual performing analysis of mission periods that are particularly hazardous, informing when extra attention should be paid to mission control and analysis. The second contribution comes from the speed at which AMSE can be performed. AMSE employs an object based modular structure for representation of the mission which allows for rapid reconfiguration of the mission to accommodate unexpected events. Additionally, the AMSE analysis can be performed fast enough that multiple options can be modeled and evaluated within a few minutes, allowing for informed decisions to be made when facing a crisis. No other methods exist that provide the unique risk informed perspective of AMSE or can be so rapidly iterated in order to find a solution to a problem. Other methods used to analyze
mission risk are generally unable to be used rapidly to conduct risk analyses during unanticipated crisis scenarios.

From the three navigational techniques described in the Previous Work section of Chapter 2, a variety of unique contributions can be identified, including: development of a general approach for risk attitude informed controls design, discovery of emergent optimal risk attitudes for operation of an autonomous system in a hazardous environment, development of a risk analysis-informed process for Control Design Integration (CDI), and the development of a method with the ability to adapt for the discrepancy between known environmental information and actual operating environment conditions. Further discussion of the PHM informed navigational control techniques can be found in Section 2.2.2 of this thesis or in previous publications by the author of this thesis [3,5,6].

The COBRA method lays the ground work for development of systems that are more adept at surviving in unknown and hazardous environments. This is enabled through contributions made to functional modelling of systems, failure analysis, system health modelling, mission success estimation, autonomous decision making, and navigation. Future development of COBRA will build on this foundation in order to make systems that are more robust and less dependent on human intervention and control.
There are several topics that must first be understood prior to discussion of the COBRA methodologies. A collection concepts critical to COBRA can be found in the sections below. Additionally, a Previous Work section is included to summarize work previously completed and published in conference proceedings. This work will be published as journal articles at a later date.

2.1 Background

COBRA builds on previous work from a variety of fields.

2.1.1 Systems and Functional Modeling

Functional modeling represents physical or information systems on a functional basis for the purpose of modeling and analysis, and is often used at early phases of design. Functional Flow Diagrams (FFD) [8] are a common tool used for the functional modeling of physical systems and the interactions between them. FFD most commonly takes the form of a network of blocks, each representing a specific system function such as generating energy or processing data, and lines representing flows between functions which can be energy, information, or physical material. FFDs often start as a high-level, black box representation that only considers flows into and out of the system. The black box is then broken down into functions representing major systems, then sub-functions representing sub-systems, sub-sub functions representing the layer below that and so on until the desired model resolution is achieved.

Function Failure Identification and Propagation (FFIP) is a tool developed for the purpose of analyzing how failures propagate through a system from function to function along flow paths until critical system failure occurs [9,10]. Failure refers to a loss of functionality, e.g., a “generate energy” function failure would mean the loss of ability to generate energy. Critical system failure refers to the loss of a function that is necessary for the system as a whole to perform the primary purpose of the system. For example, a planetary rover may experience a failure of one of its wheels and remain functional, but if a planetary rover loses its ability to transmit or receive any communication, it is considered a critical system failure. Related to FFIP is Uncoupled Failure Flow State Reasoning (UFFSR) [11] which is similar to FFIP, but instead of analyzing failures...
that follow nominal flow paths between functions, UFFSR analyzing failures that propagate through physical space. An example of such a failure is a fuel cell overheating on an Uncrewed Aerial Vehicle (UAV) and combusting, resulting in an onboard computer failure due to the fire spreading. In this case, the failure travels directly from the fuel cell to the computer and bypasses any intermediate functions or nominal flow paths. The Failure Flow Decision Function (FFDF) method [12] is a tool that analyzes failure propagation through a functional model of a system for the explicit purpose of making decisions when facing risk of functional failure in order to avoid critical system failure or to fail in the most desirable way possible. An example of this in a planetary rover is the sacrifice of a scientific instrument that has already served its primary mission function in order to protect mission-critical systems. One method for mitigating failure is to employ arrestor functions [13], which stop failure from propagating along failure paths.

2.1.2 Failure Analysis Methods

A variety of failure analysis methods are in use in industry and have been developed in academia. Methods used in industry include: Failure Modes and Effects Analysis (FMEA) [14] which has been used extensively since its development for military applications in the 1950s, Reliability Block Diagrams (RBD) [15] which is used to determine system reliability using parallel and series flow paths through blocks containing reliability data, and Probabilistic Risk Assessment (PRA) [16,17] which was developed from various industries such as aerospace, defense, and civilian nuclear power. While FMEA, RBD, and PRA can be used to do some degree of automated system design through iteration, they are limited in failure model fidelity and practicality of implementation. Several methods of failure analysis have been developed to use functional modeling methods such as FBED [17]. The Function Failure Design Method (FFDM) [18] combines FMEA and FBED to provide designers with a method of choosing specific functions and their component solutions based upon past component failure data. The Function-based Analysis of Critical Events (FACE) [19] method allows designers to modify functional models based upon critical events during a complex system’s lifecycle. The Function Failure Identification and Propagation (FFIP) [20,21] method models failure flows through FBED models. The Function Failure Reasoning (FFR) [22] method provides a simulation tool to model FFIP across a complex system. The Flow State Logic (FSL) [23] method further refines FFIP and provides a complete representation of the analyzed system’s failure state. The Uncoupled Failure Flow State Reasoner (UFFSR) [24] method identifies and analyzes failure flows that do not follow nominal flow.
pathways. Although there have been significant developments in function failure methods over the past decade, there are many areas that need to be addressed and new areas for potential innovation within the current methods to provide a complete representation of potential failures in a system.

2.1.3 Prognostics and Health Management

Prognostics and Health Management (PHM) is used to predict and prevent failures in mechatronic systems [25]. Many methods exist for PHM analysis, each with their own strengths and weaknesses, making them more or less advantageous for particular applications [10], [26]. The process of making decisions based on PHM information is referred to as Prognostic-Enabled Decision Making (PDM) [27] and can be used to decide which act presents the optimal level of risk and reward within a system. This can be an incredibly useful tool in analysis with PHM because it can be used to calculate the potential damage that could be caused to a system by one component failing.

An essential element of PHM analysis is the development of mathematic models of physical systems such as mobility, control systems, structures, or power as well as the hazards that face the systems. These models offer a prediction of the results of taking an action on the physical state of the system. The application of PHM techniques could be further extended by considering their effects on the well-being of a person in the system. In this case, a person can be treated similarly to traditional hardware with equations estimating the probability of survival based on a variety of mission-specific factors.

2.1.4 Risk Attitude

Risk attitude refers to an individual’s or organization’s preference towards acceptable levels of risk and reward [28,29]. Risk is defined as a parameter that describes the probability of a negative event occurring multiplied by the severity of the event, in the case of most physical systems damage is the primary focus of risk. Risk attitudes can vary highly between individuals and can be influenced greatly by culture, environment, work conditions, past experiences, company culture, and other factors. Risk attitudes are often represented in utility functions [30] and generally take the form of a constant that pushes the function towards a general preference. Risk attitudes can be placed along a spectrum between two general descriptors of risk tolerance and risk aversion. Risk tolerance is the preference towards higher risk in exchange for a reward. Using crossing the street as an example, a risk tolerant attitude individual is more likely to jay
walk to save a little time and get to the other side sooner. Risk aversion refers to the desire to not accept more risk in exchange for a reward. A risk averse attitude individual crossing the street will more likely go to the crosswalk and wait for the proper lights and signal to cross the street.

2.1.5 Space Mission Risk Assessment

Risk assessment for space missions can take on many different forms, each with distinct advantages and disadvantages, however they generally tend to build upon a foundation of probability modeling. Many modeling techniques attempt to represent trends of physical failure through the application various failure distributions. One common method is the use of a hazard rate, $\lambda$, which describes the expected number of failures over a period of time. The hazard rate can be used in a failure distribution such as an exponential distribution. The failure rate or a related metric appears in a wide variety of risk assessment methods, but many additional and more complex techniques exist for evaluation of risk of failure to a system. One such method for evaluating risk of failure is Failure Flow Identification and Propagation (FFIP) [22,31]. FFIP uses a functional modeling approach based in a function block diagram structure [17]. FFIP can be enhanced in order to enable mission control, navigation, and autonomous decision making through the application of Failure Flow Decision Functions (FFDF) [2]. FFDF is a tool that determines an optimal decision when faced with problems of controlling or designing a system in order to maximize system survivability. Space mission risk assessment can also be applied to control of autonomous systems in order to maximize mission success while minimizing human work hours [5,6,16,32–34].

While many of the existing methods are generally fairly robust, they suffer from lengthy setup and analysis processes. The lengthy and computationally resource-intensive setup of existing methodologies makes active assessment of dynamic situations infeasible when relying on established methods for Space Mission Risk Assessment.

2.1.6 Autonomous Mobility in Robotics

Autonomous mobility in robotic systems has a long history. Some of the earliest examples of autonomous mobile robots were William Grey Walter’s robotic tortoises, which navigated towards lights in the early 1950s and had control circuits containing vacuum tubes for decision making [35,36]. Over the past several decades, autonomous mobile robots have become significantly more advanced and now utilize advanced controls to command their complex systems.
The National Aeronautics and Space Administration (NASA) currently have two active rovers on the Martian surface: Opportunity and Curiosity [38,39]. NASA has also put two other rovers, Spirit and Sojourner, on Mars since 1997 [40]. Spirit is the twin of Opportunity sharing the same design and having been on Mars approximately the same amount of time. Spirit however failed prematurely due to becoming trapped in a dust pit on the ground and losing mobility. The dangers of unknown terrain conditions and large signal delay before human mitigating action can be initiated are a great threat in robotic planetary exploration. Curiosity is facing a similar fate as a result of unexpected damage to her wheels from unforeseen terrain conditions. The enhancement of robotics for planetary exploration through artificial intelligence and increased autonomy has a great potential and could greatly increase the breadth of human knowledge [41].

2.1.7 Route Planning

Route planning was first developed in the late 1960’s [42] and has been used widely in a variety of fields including transportation infrastructure, aerospace, automotive, and robotics [43–46]. In the field of robotics, route planning can be used in commercial applications in warehouse settings, security, tour guiding, or exploration. NASA researchers developed the OASIS autonomous science system [41] which provides a method for planning of rover scientific activities. OASIS allows for the managing of long term objectives with opportunistic scientific actions while generating a mission and route plan. Many route planning techniques involve the use of optimization techniques [47] to determine the best available path. These specific optimization objective functions [48] can vary, but generally are a non-linear constrained function that looks at the direct linear distance between two discrete points. PHM-enabled route planning has been under development for the past several years [8,49] and has shown to be effective, but has not taken into account varying risk attitudes in their decisions.

2.2 Previous Work

In addition to the two journal articles that make up Chapters 4 and 5 of this thesis, extensive previous work was performed that was published and presented at conferences. The previous work consisted of the development of the FFDF method, the creation of risk informed navigational control methods, and work towards the development of the AMSE method.
2.2.1 Failure Flow Decision Functions

The Failure Flow Decision Function (FFDF) method [12] aims to reduce the probability of total system failure, when faced with an unavoidable system failure scenario, such as operating in a hazardous environment where repair before failure is impossible. FFDF reduces the probability of critical system failure by directing the failure towards systems that are less critical to overall system survival, through the use of direct failure functions. Direct failure functions are a proposed addition to the FBED lexicon that describes functions that direct failure to a desired function. Example systems that are representative of a direct failure function include: an engineered failure point on a pressure vessel, a circuit that directs power surges, or a set of sub-systems that allow for the reorientation of the entire system to face less critical systems towards the direction of a failure initiating event.

FFDF consists of seven steps that are listed below:

1) Develop functional model of the system of interest.
2) Define critical functions and flows.
3) Assign Failure Probabilities.
4) Convert functional model into a mathematic representation.
5) Discover all failure flow paths in the system.
6) Calculate system critical failure probability from initiating failure.
7) Insert Direct Failure Function to minimize probability of critical failure.

The first four steps of the FFDF method involve the development of a functional model and creation of a mathematical representation of it. A detailed description of how this is performed is found in Chapter 4.

In Step 5 of the FFDF method all possible paths for failure propagating from every function is found. This is the most computationally intensive and difficult part of the FFDF analysis, and it only gets more complicated as the functional model gets larger. The first paper published on FFDF [12], attempted to use a Monte Carlo method to generate failure paths while performing the Step 6 calculation via Bayesian sampling, but was incredibly inefficient. The FFDF-ID case study presented in Chapter 4 uses an improved method for Step 5 that utilizes a maze solving algorithm to discover all failure paths possible in the system, and then performs the Step 6 calculation separately.
The Step 6 calculation is a form of FFIP [21] that uses all of the paths found in Step 5 and calculates the probability of failure propagation by finding the intersection of the probability that the first function in a path passes its failure along a flow line and the probability that the connected function will accept the passed failure. This calculation is performed until a critical function is failed. This process is discussed in further detail in Chapter 4. The intersection of the probability of critical system failure for each path originating from an individual function is then found, giving the total probability that failure of the individual function will lead to critical system failure. One focus of future work into FFDF will involve the inclusion of failures that do not follow functional model flows through the inclusion of UFFSR [24] analysis.

In Step 7 a direct failure function is inserted into the model. The placement of the direct failure function should allow for it to accept a failure initiating flow from the environment and direct it towards the desired functions. The flow could be representative of a physical impact, power surge, or other failure initiating event. The selected functions to be directed to should include the functions with the lowest probability of their own failure leading to critical system failure.

Further discussion of the FFDF method can be found in Chapter 4 of this thesis.

2.2.2 Risk Informed Navigational Controls

In the course of the work for this thesis, several conference papers were publish by the author on the problem of risk informed navigational controls [5,6,32]. This class of problem is of significant interest due to its representative nature of COBRA Phase 2 methods and the unique nature of navigational problems in that they take a wide variety of multidimensional data and export a two-dimensional result that is human interpretable. The primary methods of interest developed are Risk Attitude Informed Route-Planning (RAIR), PHM Informed Damage Aversion Algorithms (PIDAA), and Global to Local Path Finding Design and Operation Exploration (GLPFDOE). For each of the aforementioned methods, navigation by an autonomous Mars Exploration Rover (MER) between set target points in a simulated Martian environment was performed.

2.2.2.1 Risk Attitude Informed Route-Planning

The RAIR method [32] was the first method developed, the study of which focused on the effects of risk attitude on mission success (as defined by probability of survival and probability of
RAIR’s control algorithm works by observing the environment surrounding the rover, and then performing a calculation with an objective function in order to determine the optimal direction of travel. A flow diagram of the RAIR method is displayed in Figure 2.1.

\[
\Omega = \frac{\theta - \theta_p}{(\theta + \theta_p)/2} + \rho \frac{\Delta L - \Delta L_p - 1}{(\Delta L - \Delta L_p + 3)/2} + \int \eta \, dl
\]  \hspace{1cm}(2.1)

Making informed control decisions from this perspective, the tested RAIR rovers are able to increase their probability of survival, but this was limited by two factors.
The first factor limiting mission success is that risk attitude has a strong correlation to mission success and for overly risk tolerant or risk averse rovers, undesirable emergent behaviors develop. For the excessively risk averse rovers, they often encounter an area where they reach a local maximum in safety and mission progress. In this case, the rover turns around in circles until too much time has elapsed and the mission is halted. This behavior is very adept at increasing system survival, but undesirable due to the inability to complete the mission. In the case of overly risk tolerant rovers, they often take paths that made significant mission progress, but lead to a severely reduced probability of mission survival, and in some cases it leads to the rover becoming physically stuck in topographical features of the map. However, between these two behaviors, there are rovers that are highly successful, and accepted an appropriate level of risk to complete the mission.

The second factor that affected RAIR was being limited to only analyzing perceptible environmental hazards. Limiting the ability of the rovers to perceive all of the environmental hazards was done to increase the realism of the simulation, and had the expected result of negatively influencing mission success. In order to address this problem; a new form of control needed to be designed that responded to health effects on the system and not just perceived dangers.

Results of a case study performed on four maps for risk tolerance factor, $\rho$, of 5, 10, 20, and 50 are shown below in Figure 2.2.

2.2.2.2 PHM Informed Damage Aversion Algorithm

The PIDAA method attempts to address the inability of RAIR to properly respond to undetectable hazards, by looking at the current system health and projecting probability of mission failure from current conditions. Additionally, PIDAA is limited to not be able to observe hazards to avoid allowing for only reflexive actions to be taken when potential damage is detected. This is analogous to a person feeling their way through the dark and attempting to safely navigate through a room full of furniture. A flow diagram of the PIDAA method can be seen below in Figure 2.3.
Figure 2.2: Plots of rover paths on four maps, for risk tolerance $\rho$. YELLOW: $\rho=5$, BLUE: $\rho=10$, RED: $\rho=20$, GREEN: $\rho=50$
Figure 2.3: Process flow of the PIDAA method
PIDAA is shown to be effective at reducing the probability of system failure, however again emergent autonomous behavior complicates the results. In several cases, when PIDAA encounters a hazard it attempts to find its way around the hazard and finds that the area in one direction is safer than surrounding areas. This could lead to the rover driving off course as it continually followed a gradient of increasing perceived safety, and never returning to the desired path as shown in Figure 2.4. In order to better account for this, a method would have to be developed that allows for some planning based on observation, some reaction to system damage, and also considered a variable risk attitude adjustment that could limit the amount of autonomous damage avoidance for improved system performance.

2.2.2.3 Global to Local Path Finding Design and Operation Exploration

The GLPFDOE method [6] combines the best parts of RAIR and PIDAA while considering more complex behavior when navigating through a hazardous environment. GLPFDOE begins by taking low fidelity environmental information and generating a path plan that attempts to navigate around any known environmental hazards. During navigation through the actual environment the path is followed, but a PIDAA-based approach is used for detecting unobserved environmental hazards. When a hazard is detected, the rover attempts to use a combined RAIR and PIDAA
navigation [5] in order to navigate around the obstacle. This is kept in check through a deviation penalty that decreases the desirability of deviation from the path plan, which is representative of a risk attitude. GLPFDOE was found to be highly effective for navigation through a hazardous environment. In addition to the control design, a physical system design problem was considered. Failure mitigating functions would be placed in a functional model representative of the rover in order to reduce the probability of system failure from a class of hazard. GLPFDOE simulated navigation would be performed again and a behavior would be selected that best improved system performance. The GLPFDOE method is broken into three phases: 1) Generate path plan, 2) In-situ testing, and 3) Design iteration. A flow diagram of the GLPFDOE method is shown in Figure 2.5.

![Figure 2.5: Process flow of GLPFDOE method](image)

Eventually the system performance converged to a system with a high probability of mission success. The results of a case study of a simulated Martian rover [6] are summarized in Table 2.1. In the case study a parameter sweep of the deviation penalty, $\beta$, was performed and the $\beta$ value with the highest pre-iteration Probable Mission Completion score (pPMC) was selected. The placement of arrestor functions to reduce the probability of failure from a particular hazard were then evaluated and the arrestor function with the greatest benefit as determined by a
calculated iterated Probable Mission Completion (iPMC) score was selected. The equation for PMC is shown in Eq. 2.2 and includes the scientific objective yield of the mission run, $O_Y$, the accumulated mission hazard rate for each of $i$ hazards, $\lambda_i$, the quantity of hazard $i$ encountered, $H_i$, and the number of mission waypoints reached, $w$.

$$PMC = O_Y \cdot \left( 1 - \prod_{i} e^{\lambda_i H_i w} \right)$$  \hspace{1cm} (2.2)

**Table 2.1: Results of GLPFDOE iteration on a simulated Mars rover**

<table>
<thead>
<tr>
<th>Iterations</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\beta_{\text{min}}$</td>
<td>0.000</td>
<td>0.393</td>
<td>0.673</td>
<td>0.833</td>
<td>0.948</td>
<td>1.140</td>
</tr>
<tr>
<td>$\beta_{\text{max}}$</td>
<td>2.749</td>
<td>2.356</td>
<td>1.795</td>
<td>1.635</td>
<td>1.406</td>
<td>1.350</td>
</tr>
<tr>
<td>Mean $p$PMC</td>
<td>5.586</td>
<td>2.962</td>
<td>3.773</td>
<td>8.887</td>
<td>12.524</td>
<td>15.272</td>
</tr>
<tr>
<td>Max $p$PMC</td>
<td>8.151</td>
<td>6.352</td>
<td>8.252</td>
<td>12.856</td>
<td>18.405</td>
<td>19.975</td>
</tr>
<tr>
<td>Best $\beta$</td>
<td>1.964</td>
<td>1.234</td>
<td>1.154</td>
<td>1.406</td>
<td>1.275</td>
<td>1.290</td>
</tr>
<tr>
<td>Mean $i$PMC</td>
<td>7.708</td>
<td>0.103</td>
<td>0.182</td>
<td>1.913</td>
<td>16.113</td>
<td>19.168</td>
</tr>
<tr>
<td>$i$PMC max</td>
<td>8.545</td>
<td>0.134</td>
<td>0.208</td>
<td>2.227</td>
<td>17.391</td>
<td>20.336</td>
</tr>
<tr>
<td>Arrestor</td>
<td>8</td>
<td>8</td>
<td>2</td>
<td>1</td>
<td>7</td>
<td>3</td>
</tr>
</tbody>
</table>

GLPFDOE allows for the design of a system that is highly adapted to its operating environment from both a physical and control perspective. Furthermore, the end design is able to address some environmental hazards through controls, and harder to avoid hazards through physical design. GLPFDOE was able to outperform previous attempts with RAIR and PIDAA and perform more complex missions than had previously been attempted with multiple objectives.
CHAPTER 3

COBRA METHODS AND THEIR APPLICATION

Autonomous systems are desirable for operation in and exploration of environments that are hazardous to humans. This is due to their potential to react to unknown hazards and navigate through danger without putting human life at risk. Existing autonomous systems are limited in their capability to respond to the unknown or unexpected and are not designed with system architecture to enable effective autonomous survival decisions.

The COBRA method addresses this through a four phase approach to the design of autonomous systems that begins in the first phase with comprehensive modeling of the physical system and analysis of its theoretical health. In the second phase, this model is used to develop Prognostic and Health Management (PHM) informed, and risk-aware system controls that allow for navigation through hazardous environments and the completion of high risk tasks. In the third phase, the conceptual system model and control system design are evaluated for their effectiveness to perform a desired set of tasks making up a mission. At this point, the first two phases may be iterated until a desired probability of mission success, as determined by the third phase, is achieved. Finally, all design insights gained from the first three phases are combined into a final system design. A graphical representation of COBRA can be seen below in Figure 3.1.

![Figure 3.1: The Conceptual Object Based Risk Analysis Method](image-url)
Several methods have been developed for the implementation of each phase of the COBRA design method. Currently the most effective methods for each phase are:

**Phase 1) Modeling Analysis of System**

The Failure Flow Decision Function (FFDF) (previously published in the proceedings of the 2015 International Conference on Engineering Design) and Failure Flow Decision Function for Iterative Design (FFDF-ID) [1,12] methods, presented in Chapter 5 of this thesis, model a system using functional models [8,50] that are then analyzed to determine how failure propagates through the system. The resulting failure flows are used to determine probability of total system critical failure from smaller sub-system initiating failures of components. This knowledge can then be used to inform decisions made by the autonomous system when facing failure in order to direct failure towards sub-systems that are less critical to system survival. This fulfils the needs of the first phase of COBRA by providing a critical foundation for making informed system failure decisions.

**Phase 2) Generate PHM-Informed Controls**

Control of the autonomous system must be performed in a way that incorporates a physical understanding of the risk posed to the autonomous system informed by Phase 1, reacts to known and unknown environmental hazards, and incorporates the risk attitude of the individual or organization managing the autonomous system [29]. Three approaches to this problem are discussed in the Previous Work section of Chapter 2 of this thesis that were previously published in the conference proceedings of the 2015 American Society of Mechanical Engineers International Design Engineering Technical Conference and are accepted to the 2016 American Society of Mechanical Engineers International Design Engineering Technical Conference. The Global to Local Path Finding Design and Operation Exploration (GLPFDOE) method [6] is currently the most advanced approach to Phase 2, and achieves the goals of Phase 2 by taking an iterative approach to generating movement. GLPFDOE first looks at an entire path plan for a mission and finds the optimal path for reducing risk using initial incomplete information, such as what would be available from satellite imagery. Then as the system executes its operation it adjusts to any unknown or unexpected hazard as it is encountered and adjusts its path accordingly.
GLPFDOE exemplifies Phase 2 of COBRA by using system information from Phase 1 methods to allow for informed decisions when facing failure in order to maximize system survival. Additionally, GLPFDOE attempts to better address the problems of reacting to the unknown by performing simulated system tests with artificially low fidelity information of the environment. This is important because it allows for a control scheme to be developed that can not only respond to expected environmental hazards, but can identify and react to unexpected hazards as well.

**Phase 3) Analyze System Mission Effectiveness**

Using the physical and control systems designed in Phases 1 and 2, a simulated mission can be analyzed for probability of success. This is performed using the Active Mission Success Estimation (AMSE) method [51,52] presented in Chapter 5 of this thesis, which develops a functional model of the environment and all other systems of interest. The mission is broken into a group of small tasks that can each be individually modeled for risk towards the primary system of interest (generally the primary agent in the mission, such as a rover or a human crew). The probability for mission success can then be analyzed mathematically, and the expected probability of survival over the length of the mission can be found. Viewing the probability of mission survival as a function of time allows for both the discovery of sections of the mission that are more hazardous and should be given a greater focus, and enables decisions to be made that consider near and long term impacts on probability of mission success. One of the greatest advantages of AMSE as a tool is that it allows for the rapid analysis of missions and mission crisis scenarios. This enables to AMSE method to model decisions involving unexpected mission scenarios that have arisen with minimal time investment.

The AMSE method is exemplary of Phase 3 in that it takes a functional model as developed in Phase 1 and a control scheme as developed in Phase 2 and combines them into a greater system-of-systems for the purpose of analysis of entire mission success. This allows for the analysis of system decisions and mission frameworks for improved probability of survival and mission success.

**Phase 4) Integrate Elements into a Final Physical Design**

The conceptual system design developed in Phase 1 and the control system developed in Phase 2 and analyzed for mission success in Phase 3 is implemented into a final system design. This design is then verified through testing in a representative operating environment. Two
systems have been the focus of development for system modelling, simulation software, and hardware and are to be used for verification of COBRA methods in experimentation. These two systems are an underwater aquatic Remote Operated Vehicle (ROV) dubbed the Robotic Optimized Bathyscaphe Reliability Testbed (ROBRT) and a terrestrial rover dubbed the PHM Enabled Test Rover (PETR), which were under development over the course of the work performed for the thesis, and are currently in their fourth and second major design iteration. The current designs of ROBRT and PETR can be seen in Figure 3.2 below.

Figure 3.2: (LEFT) The 2nd, 3rd, and 4th generation ROBRT designs, (RIGHT) The 1st and 2nd generation PETR designs.
Application to design of systems for improved system survivability or improved operation when facing risk is the ultimate goal of COBRA. However, while connections between the first three phases and the fourth phase have been investigated, no formal publication has been produced on a dedicated Phase 4 method. Further discussion of this can be found in Chapter 7. While the majority of work presented in this thesis focuses on autonomous systems, COBRA has a wide variety of applications in fields such as nuclear safety, aerospace, self-driving vehicles, and infrastructure design.

The COBRA methodology has applications in space exploration, search and rescue operations, nuclear site surveying, oil field survey and maintenance, self-driving vehicles, human assistive robotic, and aerospace. Improving autonomous system reliability and ability to make informed decisions when facing risk is a critical step to enabling a wide variety of technologies that would better enable human space exploration and the capability of autonomous systems in terrestrial applications.
CHAPTER 4
FAILURE FLOW DECISION FUNCTIONS FOR ITERATIVE DESIGN

A paper to be submitted to *Journal of Engineering Design*
Ada Rhodes Short, Ann Lai, Douglas L. Van Bossuyt

Abstract

Functional modelling methods are used in early conceptual stages of design to enable the analysis and iteration of complex system designs. A family of methods exists to model functional failures and failure flows within systems, and provide an understanding of the potential system failure sources to direct redesign efforts of the system for increased reliability. These techniques are often used for analysis of systems at the early conceptual phase of system design to determine how failure may affect total system health and inform the placement of mitigation systems to prevent or reduce the magnitude of failure. However, the ability to model and analyze failure flow routing and decision-making for the purpose of functional design is not considered in existing methods. This paper presents an application of the Failure Flow Decision Function (FFDF) method for the analysis of an early design stage functional model to optimize critical sub-system survival via iterative design. Failure Flow Decision Functions for Iterative Design (FFDF-ID) enables the designer to optimize a design for the protection of critical sub-systems through the application of informed early stage analysis and design iteration. A case study based on the iterative design of a theoretical Mars Exploration Rover (MER) platform is presented in this paper to demonstrate the usefulness of FFDF-ID.

Key Words: Functional Modelling, Failure Modelling, Failure Flow Decision-Making, Design Methodology, Robust Design

4.1 Introduction

Complex systems such as rovers on Mars, underwater and airborne autonomous vehicles, and remote petrochemical installations often operate in environments where servicing or repairing damaged or broken hardware is either extremely expensive, impossible due to location of the system, or impossible due to system loss before servicing or repair can occur. Inexpensive systems such as hobbyist drones can be easily replaced but systems such as a Mars rover are much more
difficult and time consuming to replace once the system has failed. It is therefore important to deploy systems in operating environments where there is a high cost or consequence to failure that are survivable to accomplish the intended mission over the intended duration of the mission. While significant attention is often paid to understanding system failure risk and mitigating potential failures during system design, risk analysis often happens after major architectural decisions have been made and the system design has progressed to the point where system architecture changes are costly and time-consuming. Performing risk analysis early in the design process before architectural decisions have been made can contribute to designing a system that has a higher likelihood of surviving the intended mission without failure, without expensive design changes, and without compromising other aspects of the design or mission.

Efforts have been made over the last two decades to analyze risk of system failure during operation in the conceptual design phase. Groups such as NASA’s Team-X [53] have developed methods for assessing risk of system failure during conceptual mission design studies using trade-off study tools adjusted for risk analysis. Others have focused on functional system modeling as a tool to assess risk of failure. Functional modelling is a useful tool for conceptual system design that can be used to simulate a system’s responses to its environment. Iterating a functional model using simulated system response data can help to optimize a design. Several tools to analyze risk of system failure using functional models have been developed in the last decade. To date, functional failure modeling tools have focused on modeling failure risk of one functional model but have largely not been used as iterative design tools.

We present here the Failure Flow Decision Functions for Iterative Design (FFDF-ID) method that begins to realize the promise of risk assessment using functional modeling. FFDF-ID is a method of failure analysis and design iteration that uses Function Failure Identification and Propagation (FFIP) [20] and Flow State Logic (FSL) [23] to conduct failure analysis on a functional model. We recently have added a new element to the function failure analysis and design toolkit with the inclusion and use of Failure Flow Decision Functions (FFDF) that serve to channel a failure flow away from critical functions and toward functions that can be sacrificed. We previously have used FFDF to inform the execution of control decision and development of autonomous behavior in robot systems in hazardous environments [6,32]. Here we use FFDFs to iterate toward an optimized system design. Critical functions are functions that are necessary to
the core functionality of a system. The FFDFs are special functions that protect critical functions from failure flows such as in the case of an electrical failure flow that could disable a convert electrical energy to rotational energy function. Routing the electrical failure flow to a less critical function such as a convert electrical energy to optical energy function can protect the critical function. A physically realized system of the simple example could be a rover’s electric wheel motor being protected from a power surge by routing the power surge with circuit protection equipment to a light emitting diode (LED) array that is less critical to the core functionality of the rover and the rover’s mission.

While placing FFDFs in a system functional model is useful to system engineers, automating the process to automatically identify system configurations that are the least likely to fail during the mission duration is the goal of FFDF-ID. In this paper, we go beyond simple functional model iteration and use sophisticated mission analysis techniques to iterate the design using probabilistic mission simulation to determine the system designs that have the highest likelihood of surviving the mission intact and without a critical function failure. Refinement of the functional model to include design constraints such as cost, mass, or energy consumption; requiring minimum flows be produced between functions; or considering other performance characteristics such as speed of a vehicle or payload of a rocket produce more realistic assessment of potential system designs. The FFDF-ID method can assist with selection of functional solutions by zooming in on the functional model to a sub-sub-function level and identifying specific component or assembly solutions to functions by leveraging a large function-component library design repository. Additionally, FFDF-ID draws attention to systems and sub-systems that are more critical to system failure probability, and helps to inform and direct the conceptual design process.

4.1.1 Specific Contributions

This paper proposes a novel method for design optimization for system survivability informed by FFDF that uses branching iteration to optimize system survivability. This allows for rapid analysis and optimization for system survivability at the early conceptual stage of design prior to making large architectural decisions. The result of FFDF-ID is a conceptual system design that is optimized for survivability over the duration of the intended mission using FFDFs to allow the sacrifice of less critical system functions to protect critical system functionality.
4.2 Background

The Failure Flow Decision Functions for Iterative Design (FFDF-ID) method builds upon several related fields including system and functional modeling, failure analysis methods, Prognostics and Health Management (PHM) techniques, and decision theory. The pertinent topics to using FFDF-ID for design optimization are discussed in this section to provide information necessary to the development and deployment of FFDF-ID.

4.2.1 System and Functional Modeling

System and functional modeling is a suite of methods and tools for conceptualizing and developing complex systems that can be used for a wide range of analysis and design applications. A variety of techniques are used to model the functions and flows of a system, often including many sub-functions that represent physical processes such as work done in the system on flows that represent energy, material, or information moving between functions, and import and export flows that enter and exit system boundaries. Two common techniques used in systems engineering applications include Functional Flow Block Diagrams (FFDB) [54,55], and Integrated Computer Aided Manufacturing (ICAM) DEFinition for Function Modeling (IDEF0) [56]. FFBD is useful for modeling systems where there are direct linear flows between different functions and a clear system of input and output exists. However, FFBD is not as useful for modeling systems with complex flows paths where flows and functions can become difficult to model and understand, making it challenging to analyze and provide an accurate representation of the system. An alternative approach to functional modelling attempts to model the behavior and physical properties of a system via the passage of information and states through the system. Enhanced Function Flow Block Diagrams (EFFBDs) [57,58] are a variant of FFBDs that adopt this approach by modelling mechatronic control system behavior. While the EFFBD approach can offer benefits for system design, it is limited in its ability as a platform to perform system health analysis due to the lack of inclusion of functional subsystems that can represent PHM systems allowing for analysis of system health.

Other system and functional modeling techniques include Systems Modeling Language (SysML) [59] and the Functional Basis for Engineering Design (FBED) [17,60]. SysML is a flexible systems engineering-specific form of the Unified Modeling Language (UML) [61]. UML is a modeling language for general purpose applications often used in software engineering. FBED
provides concise definitions of functions and flows that can be used to develop models of possible engineered electro-mechanical systems. FBED is used in this research for several reasons including the usefulness of FBED to system engineers, the ever increasing library of failure analysis methods that are based on FBED, the well-developed FBED functional taxonomy, and existing functional component solution design repository solutions [62]. While existing system modelling tools can provide many insights into system design and help to shape system architecture before large architectural decisions are made, system modeling tools do not currently iterate or optimize designs based upon protecting critical system functions.

We have observed that the generation of functional models tends to take one of two approaches. In the first approach, a functional model is developed from an existing system through functional decomposition [63]. In this case, the system is decomposed into its sub-systems and the functionality of each system is determined. The functionality can then be represented through functional blocks and flow paths, and a functional model can be created. This method is common when developing models for analysis based on existing systems. The second approach involves the generation of the functional model from the desired functionality of the system without a pre-determined physical system design dictating the functional model. In this case, a desired system export flow may be first determined, and then the system is developed to determine necessary flow imports and system functions to achieve the desired system flow exports. The functional model can then be used for the design of the physical system through the selection of components that satisfy functions [17,60,62]. Both approaches to functional modelling allow for analysis of the system through a variety of methods, as well as enable the selection of physical components to serve a functional purpose in the system design. In this paper, we develop the initial functional model used in the case study from a known system to not distract from building an understanding of the FFDF-ID method. However, FFDF-ID works equally well with a functional model that was not pre-determined by an existing physical system.

4.2.2 Failure Analysis Methods

A variety of failure analysis methods are in use in industry and have been developed in academia. Methods used in industry include: Failure Modes and Effects Analysis (FMEA) [14] which has been used extensively since its development for military applications in the 1950s, Reliability Block Diagrams (RBD) [15] which is used to determine system reliability using parallel
and series flow paths through blocks containing reliability data, and Probabilistic Risk Assessment (PRA) [16,17] which was developed from various industries such as aerospace, defense, and civilian nuclear power. While FMEA, RBD, and PRA can be used to do some degree of automated system design through iteration, they are limited in failure model fidelity and practicality of implementation. Several methods of failure analysis have been developed to use functional modeling methods such as FBED [17]. The Function Failure Design Method (FFDM) [18] combines FMEA and FBED to provide designers with a method of choosing specific functions and their component solutions based upon past component failure data. The Function-based Analysis of Critical Events (FACE) [19] method allows designers to modify functional models based upon critical events during a complex system’s lifecycle. The Function Failure Identification and Propagation (FFIP) [20,21] method models failure flows through FBED models. The Function Failure Reasoning (FFR) [22] method provides simulation tool to model FFIP across a complex system. The Flow State Logic (FSL) [23] method further refines FFIP and provides a complete representation of the analyzed system’s failure state. The Uncoupled Failure Flow State Reasoner (UFFSR) [24] method identifies and analyzes failure flows that do not follow nominal flow pathways. Although there have been significant developments in function failure methods over the past decade, there are many areas that need to be addressed and new areas for potential innovation within the current methods to provide a complete representation of potential failures in a system.

4.2.3 Hazard Rate Estimation and Success Assessment

One way of analyzing the likelihood of a system completing its intended mission is assessing the system’s hazard rate. The hazard rate is typically based on the probabilistic assessment of system failure. Estimating risk using a hazard rate, \( \lambda \), can be achieved with an exponential distribution, Eq. (4.1), to calculate the expected survivability rate at time, \( t \), [64].

\[
S = e^{-\lambda t}
\]  

The expected survival equation can be derived to find the expected failure rate function (F), Eq. (4.2). This can be done by subtracting the probability of survivability, S, survivability from 1.

\[
F = 1 - e^{-\lambda t}
\]  

30
Under this convention, survivability rate, $S$, is the probability that a system will be functional at a given time and Failure Rate, $F$, is the probability that a system will not be functional at a given time. Modelling of failure rate and survival rate are very critical to construction of the FFDF model for use in FFDF-ID.

4.2.4 Failure Flow Decision Functions

The FFDF method enables the making of informed failure mitigating decisions when designing or managing complex systems [12]. FFDF beings with development of a functional model to represent a system of interest. The functional model is then enhanced with models for the passing of failure for every function and flow. The models include the probability that a flow will carry a failure from a failed function and the probability that a function will accept a failure from a flow. Critical functions and flows in the model are then defined (see Step 2 of the Methodology section for more detail on how this is performed). Using the mathematical models of failure, FFIP analysis can then be performed on the functional model in order to determine the probability of failure of critical systems. The provided insight from analysis can then be used to inform system decisions when facing failure. One example application of FFDF is the analysis of power grids for informed decisions under impending failure, such as if power must be cut to one section of the power grid. FFDF can provide insight on how cutting power to that section of the grid affects other sections of infrastructure. FFDF is integral to the FFDF-ID method and where portions of the FFDF method are found in Steps 1 through 6 in the Methodology section below. FFDF-ID builds upon the foundation of FFDF by shifting the problem focus from system operation and controls and into design of a system for improved survivability when applying FFDF for control and management decisions.

The FFDF-ID method builds upon previous work in system and functional modeling, failure analysis methods, prognostics and health management, hazard rate estimation and success assessment, and previous development of the FFDF method. FFDF-ID applies these concepts in order to enable the design of systems for improved survivability, and enhances the effectiveness of the design for use with FFDF for control and management decisions when facing failure.
4.3 Methodology

The Failure Flow Decision Function for Iterative Design (FFDF-ID) method presented here is a tool that uses functional modeling, failure analysis, and iterative design to develop conceptual designs of complex systems that are more likely to maintain critical functionality over the duration of a mission. FFDF-ID consists of eleven steps organized into three distinct phases that are performed using computational tools to optimize the final functional model design. Figure 4.1 shows the eleven steps and iterative loop of FFDF-ID. The FFDF-ID method begins with development of a representative functional model of the system using FBED’s functional modeling taxonomy (Phase 1). The functional model is then analyzed using functional failure analysis techniques such as FFIP to determine the projected survivability of the system (Steps 5-7 in Phase 2). In this paper, we only use the base FFIP analysis method for clarity although FFDF-ID works equally well with other functional failure analysis techniques such as Uncoupled Failure Flow State Reasoning (UFFSR) [24]. The functional model is then modified within the constraints of the design problem (Step 8-10 in Phase 2). Functional failure analysis is run again on the revised functional model to determine whether survivability has improved by the placement of FFDFs (Steps 5-10 in Phase 2) before the design is iterated again. FFDF-ID is targeted at the early conceptual stage of system design prior to making large architectural decisions. While FFDF-ID can be used to develop designs with a higher probability of survival for the duration of the intended mission, FFDF-ID is particularly useful for systems that will be used in environments where repair is impractical or impossible.

Phase 1: Generate Functional Model

Step 1: Develop Functional Model

The first step in FFDF-ID is the development of a functional model (we recommend using the FBED taxonomy [60]) of the system of interest that provides a system model for later analysis. High-level functions are selected to represent all major functions of the system being modelled. The functions are then connected with flows, as shown in Figure 4.2, where dotted lines represent signals or information being passed between functions, solid thin lines represent energy flows such as electricity or heat, and thick solid lines represent the passing of physical material. Figure 4.2 is a simplified functional model of a Mars rover. The functional model used in the case study below is derived from the model presented in Figure 4.2 and presented in Appendix A.1.
Figure 4.1: Process Flow of FFDF-ID
Step 2: Define Critical Functions and Flows

The next step in FFDF-ID is identifying critical functions and flows. Others have defined critical functions as a set of design functions observed in a single domain that significantly define the functionality of a system [65]. We modify the definition of critical functions and flows in this paper as follows: critical functions and flows are elements of the functional model that must perform their intended functions or flow nominally for the system to not be in a failure state. Further, we divide critical functions and flows into two major classes that allow for numerical and probabilistic analysis of the function model. The two classes are defined as: 1) independent critical functions (ICFs), and 2) $k$ of $N$ critical functions (kNCFs). ICFs are single functions that if failed...
lead to the entire system losing functionality. An example of ICFs for the case of a Mars rover is the CPU in the form of a Process Signal function. The Process Signal function failing leads to the rover not performing any computational tasks including receiving data or controlling subsystems which causes the rover to no longer operate even if the rest of the rover is in perfect working condition. Thus, Process Signal is be defined as a critical function in the case of a Mars rover. The second class of critical functions, kNCFs, are sets of functions of size N of which some quantity represented by the integer value k must be functioning nominally for the system to not be in a failure state. An example of this from a Mars rover is a solar array in the form of multiple Accumulate Energy functions. If k individual Accumulate Energy functions are functioning optimally of N total Accumulate Energy functions, then the system is not put into a failure state. Typically, the set of critical functions for a functional model of a system will consist of multiple ICFs and kNCFs.

Step 3: Assign Failure Probabilities

The third step in FFDF-ID is developing failure probabilities for each function and flow in the functional model using the methods developed in FFIP, FSL, and FFR [21–23]. The assigned failure probabilities include the probability that a function will fail, the probability that the failure of a functional will be passed out of the function along a flow path, and the probability that a function on the other end of the flow path will accept the failure. By taking the union of the probabilities of failure, passing the failure, and acceptance of the failure, the probability of failure propagating along a specific flow path can be calculated. The selection of appropriate probabilities is one of the most critical components to the development of a system model that is consistent with reality. Representative generic failure probabilities can be found in various sources appropriate to the type of system being designed (for instance, the probabilities used in the case study presented in this paper are based on values derived from [66] which are representative of Mars rovers and other space systems), gathered experimental from laboratory testing, or derived from similar systems already in operation (such as in the case with PRA analysis of conceptual designs of nuclear power plants [67,68]).
A database is then developed to store the failure probabilities. We advocate implementing the failure probability database in a way that is efficiently machine-searchable and that can be easily revised as models are expanded and updated. Appendix A.2 provides an example of the failure probability database that is used in the case study in this paper.

**Step 4: Convert Functional Model into a Mathematic Representation**

Before analysis of the functional model can begin in the FFDF-ID method, the functional model must be converted from a graphical representation (a graphical representation the most common way for functional models to be presented for human readability) to a machine-readable mathematic representation. Previous work exists for conversion of functional model graphs into a machine-readable form [12,69]. In FFDF-ID, we represent functional models through a cell array with each element in the cell array containing a string and two matrices. The string contains the type of function represented by the cell. The first matrix contains all of the flows out of the function in the first column and all of the functions that the flows terminate at the function in the second column. This allows for the tracking of failure flow during the iterating and analysis phases of FFDF-ID. The cell array thus contains the entire functional model in a machine-readable format that is computationally efficient for FFDF-ID. An example subsection of a functional model of a Mars rover is displayed below in Model 1.

### Model 1: Sample Functional Model

<table>
<thead>
<tr>
<th>Function</th>
<th>Flows In</th>
<th>Flows Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>'Accumulate Energy'</td>
<td>[f0 1]</td>
<td>[f1 18]</td>
</tr>
<tr>
<td>'Store Energy'</td>
<td>[f1 15]</td>
<td>[f1 15]</td>
</tr>
<tr>
<td>'Convert Rotation to Translation'</td>
<td>[f8 30]</td>
<td>[f7 1]</td>
</tr>
<tr>
<td></td>
<td>[f6 21]</td>
<td></td>
</tr>
</tbody>
</table>

**Phase 2: Design Iteration**

**Step 5: Define All Flow Paths for Machine Readability**

An interesting feature of functional modeling is the way that flow paths can propagate failures in parallel and can at times also propagate failures in the opposite direction of a flow’s nominal flow path. In the physical world, a component-level example of parallel failure flow paths an over-voltage failure flowing along multiple parallel wires to multiple electrical devices. An example of a failure flow that can propagate in the reverse direction of a nominal flow path at the
component level is a water hammer through piping. These phenomena in functional modeling are important to FFDF-ID because of the need to automatically compute failure probabilities of many functional models. For the purposes of FFDF-ID analysis, we have found that a functional model is best represented as a specialized network of nodes. Even a small functional model with a small network of nodes can become very complex due to parallel and reversed failure flows. Due to the increased complexity of the network representing the functional model, many traditional methods for solving possible failure flow paths are insufficient due to the exceptionally large computational resources needed, creating an interesting computational problem.

We have adopted and modified a biased wall following maze solving algorithm [70] to navigate through the network representing a functional model while recording the path as it is traversed. In our implementation of FFDF-ID, the functions and all the connecting paths are ordered numerically by their position in the functional model (see Step 4 and Model 1 for an example). The lowest ordered function in the functional model is navigated to via the lowest ordered path. A function cannot be navigated to if it has already been “visited” by the computational tool on the current path. When no more paths are available, a step is taken backwards to the last function that has viable paths that have not yet been taken and the process is continued. When no more “legal” paths (paths that are possible, as defined by the machine-readable functional model) exist then the mapping of flow paths (the functional model flow path solving algorithm) is complete. A flow chart of the process to identify all potential failure flow paths is shown below in Figure 4.3. The algorithm has been successfully implemented in software and is used in the case study in this paper.

The functional model flow path solving algorithm derived from a biased wall following algorithm that we use in FFDF-ID is preferable to other network solving methods because it is computationally simple, relatively fast, and exhaustive of all possible paths as compared to other techniques such as the use of a Monte Carlos method to generate paths through Bayesian sampling which is slow [12], topological sorting which was infeasible due to inclusion of multidirectional flows [71], or a brute force method which generates a list of all potential paths ignoring flow logic and then checks them against a model, which was feasible for small models but required prohibitive quantities of memory for larger models due to factorial growth of the memory required.
Figure 4.3: Process for solving functional model failure flow path problem
Step 6: Calculate System Critical Failure Probability from Initiating Events

The purpose of this step in FFDF-ID is to determine how the failure of each individual function in a system’s functional model can lead to critical system failure through propagation of failure flows throughout the system. For the purpose of this step in the FFDF-ID analysis, it is assumed that the initiating failure event has occurred and therefore the probability of initiating event failing is set to 1 representing a 100% chance that failure will occur.

We develop an algorithm below that steps through each failure flow path beginning with the selected initiating event. If the failure flow path contains no critical functions, then it is disregarded and the next failure flow path is selected. If the failure flow path contains an independent critical function, or ICF. If the failure flow path contains an ICF then the probability that the ICF will experience a failure, $F_{ICF}^p$, is calculated by taking the intersection of all probabilities of events that must occur to fail the ICF. Each event in the path that must occur is made up of two probabilities, $F_{Pass}^{pf_{i+1},pf_{i}}$, the probability that a function in position $i$ in the flow path $pf$ passes failure to its neighbors, and $F_{accept}^{pf_{i+1},pf_{i}}$, which is the probability that function in position $i+1$ of the path accepts a failure passed from a neighbor. This is performed for all ICFs in the failure flow path and can be seen in Equation 4.3 in Formulation 1 shown below.

Next the algorithm checks if a kNCF is in the failure flow path. If a kNCF is in the path, then the algorithm calculates the probability of failure for each kNCF function in the path and records the value. This value is used later in the algorithm to calculate $F_{kNCF}^p$ which is the probability of failure for all kNCF sets propagating from an initiating event. A mathematical representation of this process is shown Eq. 4.4 in Formulation 1 below.

The probability of each ICF and kNCF is then recorded before the algorithm moves to the next failure flow path. After all failure flow paths for an initiating event have been examined, the union of all probabilities of critical failure is taken. Taking the union allows for the algorithm to account for all possible outcomes that lead to critical failure, and is shown in Equation 4.5. The total probability of critical failure for the initiating event, $F_{CritFail}^c$, is then recorded and the algorithm selects the next initiating event by identifying the next function that may fail independent of other failure events in the functional model to be the new initiating event and repeats the process until all initiating events are accounted for. A flow chart of the process can be
seen in Figure 4.4, and a formulation for calculating the probability of critical system failure from an initiating failure is shown below in Formulation 1.

*Formulation 1: \( F_{\text{CritFall}} \) Formulation of Critical System Failure for from initiating failure event*

Here we provide the mathematical formulation for \( F_{\text{PathCrit}} \), the probability of critical system failure for a path.

**Formulation 1.1: Sets**

- \( f \in M \): Set of all function types in the functional model
- \( f \in ICF \subseteq M \): Set of independent critical functions
- \( f \in kNCF_t \subseteq M \): Set of all function in \( k \) of \( N \) critical functions set of type \( t \)
- \( t \in T \): Set of all types of \( kNCF \) functions
- \( \varphi \in V \): Set of all flow types in the functional model
- \( p_{i,c} \in P \): Set of all possible paths in the model, containing \( i \) rows and class, \( c \), denoting class. The first row contains all functions in order along the length of the path, and the second row contains all flows in order.

\[
\begin{align*}
p_{i,c} &= \{ f_1, \ldots, f_{i-1}, f_i \} \\
p_{i,c} &= \{ \varphi_1, \ldots, \varphi_{i-1}, 0 \}
\end{align*}
\]

- \( c \in C \): Set of classes of elements in a path, containing functions, \( f \), and flows, \( \varphi \)
- \( p_{f_{i,c}} \in UP_f \subseteq P \): Set of all unique paths originating at function \( f \) and ending a \( ICF \) or \( kNCF \)
- \( p_{f_{i,c}} \in IE \subseteq P \): A set of all paths originating from a function of interest for receiving directed failure flows

**Formulation 1.2: Parameters**

- \( F_{f,\varphi}^\text{pass} \): The probability that the function, \( f \), will pass a failure along path, \( \varphi \)
- \( F_{f,\varphi}^\text{accept} \): The probability that function, \( f \), will accept a failure passed along path, \( \varphi \)
Formulation 1.3: Calculation

If a flow path originating from a function \( f \) contains no ICFs of kNCFs the probability of critical system failure is given by.

\[
\{ p_{f_{i,c}} \in UP^f | p_{f_{i,c}} \notin ICF \cup kNCF \} \rightarrow F_p = 0
\]

If a flow path originating from a function \( f \) contains an ICF, then the probability of critical system failure is given by

\[
\{ p_{f_{i,c}} \in UP^f | \forall i \neq 1, p_{f_{i,c}} \in ICF \} \rightarrow F^{ICF}_p = \prod_{i \in p_{f_{i,j}}} P^{pass}_{p_{f_{i,j}}p_{f_{i,p}}} \cdot F^{accept}_{p_{f_{i+1,j}}p_{f_{i,p}}}
\]

(4.3)

Probability of failure from kNCFs must be calculated for all paths containing kNCFs originating from function \( f \) at the once, and is given by,

\[
\{ p_{f_{i,c}} \in UP^f | \forall i \neq 1, p_{f_{i,c}} \in kNCF \} \rightarrow F^{kNCF}_p = 1 - \prod_{kNCF \in T} \left( 1 - \prod_{f \in kNCF} \left( 1 - \prod_{i \in p_{f_{i,j}}} (P^{pass}_{p_{f_{i,j}}p_{f_{i,p}}} \cdot F^{accept}_{p_{f_{i+1,j}}p_{f_{i,p}}}))) \right)
\]

(4.4)

The combined probability of failure for all paths originating from a function \( f \) is given by

\[
F^{CritFail}_{IE} = F^{kNCF}_{p_f} \cup \bigcap_{p_{f_{i,j}} \in IE} F^{kNCF}_{p_f}
\]

(4.5)

Step 7: Determine if Desired Level of Survivability Has Been Achieved

Using the results of the calculations performed in Step 6, determine if a desired level of system survivability has been achieved. The metrics of interest for this analysis of the viability of a design for survivability in FFDF-ID are the undirected and directed probability of critical system function or flow failures, including kNCFs and ICFs. A directed failure is the result of inserting an FFDF [12] function into a system in order to direct system failures away from high probability of failure paths and towards lower probability of failure paths (see section 2.6). \( F_i \) denotes probability of critical system failure for the system if failure is directed to system \( i \). By directing failure in this way a systems probability of survival can be significantly improved. The undirected
failure probability is a failure that is randomly assigned to any function in the functional model with equal probability. The probability of undirected failure, $F_u$, can be found by summing the probability of each individual function leading to critical failure multiplied by the probability that failure is directed to the function, which in the uniform undirected case is equal to one divided by the number of functions in the model, $n_f$. Mathematically this is equivalent to the mean of all $F_i$. The equation for probability of undirected failure can be found in Eq. 4.6.

$$F_u = \sum_{i=1}^{n_f} \frac{F_i}{n_f} \quad (4.6)$$

![Flowchart](image.png)

*Figure 4.4: Process for Calculating total failure probability for an initiating event*
An alternative but equally valid implementation the probability of survival. This can be done by simply subtracting the probability of failure from 1. In this case $F_i$ and $F_u$ and replaced with $S_i$ and $S_u$ respectively.

$$S_u = 1 - F_u = 1 - \sum_{i=1}^{n_f} \frac{F_i}{n_f} = \sum_{i=1}^{l=n_f} \frac{S_i}{n_f} \quad (4.7)$$

The uniform undirected failure probability is useful when external initiating event probabilities are not yet known. For instance, the next generation of autonomous space exploration systems is expected to explore regions of icy moons and outer planets that are poorly understood and where highly dynamic environments may exist. The uniform undirected failure probability of critical function or flow failures is of interest for such scenarios because it is mathematically equivalent to the case that all initiating events are equally probable and calculating the probability of failure of the entire system as a result. The assumption of equal probabilities of initiating events is an acceptable practice when making decisions under high uncertainty where external initiating event probabilities are not yet known, which allows for so that a comparison of system architectural options using risk analysis information where insufficient information exists to accurately model initiating event probabilities can be made. This is a necessary acceptable assumption in cases where the rover system is expected to be operating in an unknown or poorly understood environment where probabilities of external initiating events cannot yet be feasibly determined. In cases where the operating environment is better understood and probabilities can be assigned for individual initiating events occurring, the weighted probability of critical system failure should be calculated instead of the mean. Calculating the weighted probability can be achieved by multiplying the probability of failure propagation down a path for an initiating event leading to critical failure by the probability that an initiating event occurs. The total system failure probability can then be found by calculating the combined failure probability of all initiating events. The probability of total system failure should then be compared to a pre-establish level of acceptable risk of probability of failure or desired probability of survival. In practice the more iterations that are performed the higher the survivability of the system should become. However, as more iterations are performed and configurations are generated more computational resources are required. If the desired level of survivability is reached then the process skips ahead to Step 11: Interpretation of Results, if the desired level is not reached then continue to Step 8: Select Existing Configuration with Highest Survivability.
Step 8: Select Existing Configuration with Highest Level of Survivability

From the current set of all existing system configurations (on the first iteration there will be only one configuration), select the configuration that has the highest level of survivability as determined by Steps 6 and 7. This is the configuration that will be iterated on in order to further improve the design.

Step 9: Define Allowable Design Changes with Constraints of the Design Problem

Taking the model selected in Step 8, look for any components or configurations in the system that can be changed without violating any design constraints put on the system. Examples of design constraints include: system must be solar powered, system must be below some weight, system has a set budget, system must contain at least some number of a component, and most important of all system must be able to perform intended function. Allowable changes are any changes to functional model representation of the design that do not violate the design problem constraint. Examples of allowable changes include: replace a sub-system with a different sub-system that serves the same function, change the number of a particular type of function, or adjust flows between functions. The selected allowable change will then be used in Step 10 to generate variant functional models.

Step 10: Generate Variant Functional Models to Represent Options in the Design Space

Taking the selected allowable change from Step 9, modify the functional model in order to generate a variety of functional models, essentially performing a parameter sweep in the design space. Once the modified functional models have been generated, the process can return to step 5 in order to calculate the new configurations’ probability of failure from initiating events and continue the process from that point. Steps 5-10 are performed on the system design as an iterative loop, selecting the best iterations for further optimization. The FFDF-ID method is paired with an evolutionary algorithm to allow for many iterations to be run automatically and a design to be selected that is optimal for survival. In cases where the goal is to explore design space, varying the number of a specific functions in parallel in order to determine the sensitivity of the system of interest to addition or subtraction of redundant functions. A graphical representation of the FFDF-ID can be seen above in Figure 4.1.
Phase 3: Interpret and Report Results

Step 11: Interpretation of Results

The FFDF-IDs probability of system survival is dependent on where failure is directed through use of a FFDF [12]. Failure should be directed to whatever system has the lowest probability of leading to critical system failure. This is in-line with base FFDF methodology. In the iterative design phase of FFDF-ID analysis should be performed to decrease the probability of system failure when failure is directed through FFDF first to increase the probability of system survival as much as possible. If the probability of directed failure cannot be improved any further, continued iteration can be performed to improve the probability of undirected failure.

Due to FFDF-IDs basis in functional modelling it is important to properly understand the results of analysis and its proper application to systems. The biggest part of this is understanding what aspects of the design are being analyzed and how that relates to an actual physical system. For example, if the number of a particular type of function directing a flow into a function is adjusted and this has a benefit to the design, it does not indicate that the only way to increase system survivability is to include physical components representative of that function in a higher number. Instead it may be better to select a single physical component that has a higher probability of survival than the group of modelled functions, this would have a similar effect on the survival of the system. Additionally, a functions influence on total system survivability cannot be assumed to apply to all physical components that are capable of completing the functional need. It is incredibly important that any physical components chosen for design be validated for failure properties and verified to meet the model failure specifications. Due to the iterative nature of FFDF-ID perceived design incites should be checked by reconfiguring the model to align with the new design, and analysis can be run again, in order to check that assumptions made about effects to the total system health from small changes are accurate. Finally, it is critical that interpretation of FFDF-ID acknowledge that the results of analysis are limited by the accuracy of the model to the system of interest. Therefore, it is very important to pay attention to flow paths, function block selection, and the resolution of the model to insure that incite gained is valuable and representative of reality.
The FFDF-ID method can be applied to a wide variety of systems in order to improve the survival of the system when facing failure. Survival of systems is highly dependent on the health of a few components that are more highly linked to critical system failure but through FFDF sub-systems can be found that can be sacrificed with minimal effect on system survival. The first seven steps of the FFDF-ID methodology describe the base FFDF methodology. FFDF-ID builds on this foundation by added capability to iterate the system design for improved survivability.

4.4 Case Study

A case study is performed in which a functional model of a simplified Mars rover resembling Sojourner [72] is analyzed. The functional model starts with a base design that trades kNCFs with each other in order to optimize the system design. Structuring the case study this way as a simplified integer optimization problem is representative of a common class of design problems [73].

4.1 Case Study

In this section, we present a case study in which a functional model of a simplified Mars rover resembling Sojourner [31] is analyzed. The functional model starts with a base design that trades kNCFs with each other in order to optimize the system design. Structuring the case study this way as a simplified integer optimization problem is representative of a common class of design problems [36].

Step 1: Develop Functional Model

The functional model is limited to 50 functions and derived from the Sojourner system design. A total of 46 functions were used in the construction of the model, and a list of the functions used and the sub-systems that they represent can be found in Table 4.1. The functional model is shown in Appendix A.1. The functional model used in this case study is developed to be a simplified representation of the systems of the Mars rover Sojourner and is sufficiently complex to demonstrate FFDF-ID while being sufficiently simple to be easily understood. The layout of the model is a more detailed version of a model used in previous papers [6,12,32].

Step 2: Define Critical Functions and Flows

The functional model includes 2 ICFs and 7 kNCFs. The two ICFs are the process signal function, representing the CPU, and the transmit data function, representing the communication systems. These two functions are chosen because if either of them is lost, the rover will cease to
communicate with human controllers on Earth. The kNCFs includes the accumulate energy functions representing the solar array, the store energy functions representing batteries, the store data functions representing on-board memory modules, the record visual functions representing cameras, and the convert rotation to translation functions representing wheels. These functions are kNCFs because they represent capabilities of the rover that are necessary to perform its mission of exploration but that have sufficient redundancies to allow for continued functionality of the system with loss of some of the functions.

Table 4.1: Functions Used

<table>
<thead>
<tr>
<th>Sub-System</th>
<th>Representative Function</th>
<th>Quantity Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accumulate Energy</td>
<td>Solar Arrays</td>
<td>13</td>
</tr>
<tr>
<td>Store Energy</td>
<td>Batteries</td>
<td>3</td>
</tr>
<tr>
<td>Distribute Energy</td>
<td>Power Control Board</td>
<td>1</td>
</tr>
<tr>
<td>Direct Command</td>
<td>Primary CPU</td>
<td>1</td>
</tr>
<tr>
<td>Process Signal</td>
<td>Primary CPU</td>
<td>1</td>
</tr>
<tr>
<td>Store Data</td>
<td>Computer Memory</td>
<td>3</td>
</tr>
<tr>
<td>Record Position</td>
<td>Positional Sensors</td>
<td>1</td>
</tr>
<tr>
<td>Record Visual</td>
<td>Cameras</td>
<td>3</td>
</tr>
<tr>
<td>Process Signal</td>
<td>Communication Systems</td>
<td>1</td>
</tr>
<tr>
<td>Transmit Data</td>
<td>Communication Systems</td>
<td>1</td>
</tr>
<tr>
<td>Control Magnitude Electrical</td>
<td>Motor Control Board</td>
<td>1</td>
</tr>
<tr>
<td>Convert Electrical to Rotation</td>
<td>Driven Motors in Rocker Boagie Suspension</td>
<td>10</td>
</tr>
<tr>
<td>Transmit Rotation</td>
<td>Steering Columns on Wheels</td>
<td>4</td>
</tr>
<tr>
<td>Convert Rotation to Translation</td>
<td>Wheels</td>
<td>6</td>
</tr>
</tbody>
</table>

Step 3: Assign Failure Probabilities

Failure probabilities are derived from [66] in this case study are representative but are intentionally modified to not be real failure data. Further, we explicitly state that the failure probabilities and results of the analysis presented here, while being very useful for demonstration of the FFDF-ID method, are not intended for use in any one specific design. Practitioners must develop their own failure probability data for their specific applications and cannot rely upon the data presented here. A complete list of the failure probabilities used in this case study can be found in Appendix A.2.

Step 4: Convert Functional Model into Mathematic Representation

The functional model is then transformed from its graphical representation into a machine-readable mathematical representation using the format described in Step 4 in the Methodology.
Converting a graphically represented functional model to a mathematical representation is currently achieved via a manual process and is labor intensive. However, manual conversion still allows for the relatively rapid generation of functional models compared to our previous work [12]. In the future, we expect to develop an automated tool that converts back and forth between graphical and mathematical representations of functional models with ease.

**Step 5: Define All Flow Paths for Machine Readability**

All possible flow paths in the functional model are then calculated using the functional model flow path solving algorithm. We have found this algorithm to be computationally inexpensive and our software implementation is able to perform rapidly on large, complex functional models. In the rover model used for the case study 1371 paths were found and analyzed.

**Step 6: Calculate System Critical Failure Probability from Initiating Events**

The critical failures are calculated using the generated failure flow paths as described in Step 6 of the methodology section. This is done by first determining if a path contains an ICF or at least k kNCFs from the same set. If a flow path meets this condition, then the probability that a critical failure occurs is calculated using equation 4.3 through 4.5 from Step 6 of the methodology. Using a relatively short path generated in Step 5 for the rover functional model as sample calculation is performed. In this case the failure initiates at a Store Energy function representing a battery, potential causes of such a failure could include a short across the battery, overheating, or mechanical failure due to physical impact. The failure propagates from the battery to the distribute electricity function representing a power control board, which accepts the failure. From there the failure is passed to a direct command function representing a part of the primary CPU. Finally, the failure is passed to the process signal function representing the CPUs processor and the rover experiences critical failure. Using Eq. 4.3 values shown in Appendix A.2 the calculation is performed and shown as Eq. 4.8.

\[
P_{ICF}^{pf} = \prod_{i \in p_{f_{ij}}} F_{pass}^{p_{f_{ij}}} F_{accept}^{p_{f_{ij}} p_{f_{ij}}} p_{f_{ij}} p_{f_{ij}} = 0.8 \times 0.24 \times 0.8 \times 0.44 \times 0.99 \times 0.01 = 6.69 e^{-4} \text{ [PSD]}
\]

Note that is individual path has a very low probability of occurrence, as many paths will, but when combined with the other 1371 paths the probability of system failure can become quite large. It should also be noted that in this example all downstream paths were taken, however failure
Step 7: Determine if Desired Level of Survivability of Critical Functions and Flow Paths is Achieved

Calculate the directed failure and undirected failure probabilities for causing critical system failure as described in Step 7 of the methodology. The level of survivability is reported in terms of Probability of Survival on Demand (PSD), which defines the probability that a system does not experience failure in response to a demand. In the context of FFDF and FFDF-ID, PSD is the probability that critical system failure will not occur as a result of the system encountering a failure initiating event. For example, if a critical system failure is experience 1 out of 4 times that a system is dropped, it would have a 0.75 PSD, or in other words, it would survive 75% of the time. A few example directed failure probabilities include 0.0722 PSD for failures directed to a collect energy function representing a solar cell, 0.0444 PSD for the distribute energy function representative of a power control board, 0.000031 PSD for a failure directed to a convert electrical to rotation function representing a motor in the drive train, and 1 for failures directed to the process signal function representing the CPU (which is itself a critical function). The undirected failure probability of survival can be found by taking the mean of all of the directed failure probabilities of system survival as described in Equation 4.6, and is found to be 0.0292 PSD for the initial system design. These values should then be compared to predetermined values for desired level of survivability. For the purpose of this case study the desired level of survivability is 0.75 PSD for directed failure to a solar panel and 0.3 PSD for undirected failure, however the desired level does not have to include both metrics and could just involve the undirected failure or directed failure to several functions. Additionally, this case study will consider failures directed towards batteries, cameras, and computer memory as secondary basis of comparison. A final tie breaker of minimized changes from the initial design was used if the previous metric did not provide a definitive direction. These we selected because they have relevance to the allowable changes in this particular case study. The level of resolution in the desired level of survivability is dependent on the case study and the risk attitude [74] of the designer. If the desired level of survivability has been reached then by any configuration then skip to Step 11, otherwise continue to Step 8. In this case two iterations were needed to reach the desired level.
Step 8: Select Existing Configuration with Highest Survivability for Iteration

On the first iteration this step simply involves selecting the initial system configuration. However, on later iterations the selected system configuration should be the system that is closest to the desired level of survivability. In the unlikely case of a tie attempt to resolve it through additional objectives such as directed failure probabilities of other functions or secondary heuristics such as cost or weight.

Step 9: Define Allowable Design Changes within Constraints of the Problem

Defining allowable design changes is one of the most difficult steps of the methodology to implement because of how open ended it is. It starts to first define what can’t be done in form of constraints on the system, this helps to narrow possible options. Next attempt to select the most pertinent design change to the system (though all changes will have an effect on system health and can be studied if desired). In this case the allowable design changes are constrained by system cost, weight, volume, and functionality to include design problems. The first is a trade off in the number of accumulate energy functions representing solar cells and the number of store energy functions representing batteries and the second is a trade off in amount of data storage functions representing memory on board and record visual functions representing cameras. These problems are of particular interest because they also all involve changing the number of functions in kNCF sets.

Step 10: Generate Variant Functional Models to Represent Options in the Design Space

Variant functional models should be generated in a way that performs a parameter sweep across the design space. In the case of the first iteration looking at a trade-off between accumulate and store energy functions, four viable variations were found in addition to the control configuration. The four configurations were 7 accumulate energy functions and 5 store energy functions, 10 accumulate energy functions and 4 store energy functions, 16 accumulate energy functions and 2 store energy functions, and 19 accumulate energy and 1 store energy functions. For the second iteration a trade-off between record visual information and store data functions was studies. In this case the second iteration allowable design changes were not restricted by the first iteration (which will not always be the case), so the second iteration changes will remain the same regardless of the results of the first iteration. Again five configurations were found including a configuration representing the previous design. In addition to the 3 record visual functions and 3 store data functions of the previous iteration there is also an allowable 5 record visual functions
and 1 store data function, 4 record visual functions and 2 store data functions, 2 record visual functions and 4 store data functions, and 1 record visual function and 5 store data functions configurations.

**Step 11: Interpret Results for Design Insights**

After all early steps have been performed and desired level of survivability has been achieved the results must be interpreted in a way that is useful to the design of the system. One of the greatest insights to be gained from FFDF-ID is the optimal placement of an FFDF function into a design in order to direct system failure [12] to maximize probability of survival. This is especially impactful in FFDF-ID by coupling optimizing of the quantity of redundant subsystems with optimizing FFDF placement. The case study’s outcome is reported in the Results and Discussion sections below.

**4.5 Results**

The results of the case study demonstrate the viability of FFDF-ID as a method for iterative design optimization when facing high risk failure scenarios. We present the results here as survivability statistics rather than failure statistics to make clear that we are interested in the critical functions and flows being maintained rather than interested in any one part of the system failing. Further, survivability is an important metric for systems that are operating in conditions where repair is difficult, undesirable, or impossible, and time before repair or system loss needs to be maximized. This is especially pertinent for systems used in space exploration where repair is impossible such as the case for rovers on Mars.

The range of possible accumulate energy function combinations is examined from the first iteration through Step 10 of the FFDF-ID method. The designs from the first iteration include system configurations with seven accumulate energy and store energy functions; ten accumulate energy and four store energy functions; sixteen accumulate energy and two store energy functions; and nineteen accumulate energy and one store energy function(s). The four functional models are then compared to the original functional model derived from the Sojourner design. The original functional model has a survivability rate of 0.0722 PSD which serves as the baseline for comparison of future iterations of the functional model. The highest survivability rate is 0.218 PSD and is found in both the seven and ten accumulate energy configurations. Additionally, the secondary metrics of directed failures to other systems of interest do not offer a consensus on
improvement to survivability. The functional model with ten accumulate energy and four store energy functions is then selected to continue to the next round of analysis because this functional model represents the least amount of change from a human-generated design. We choose this approach because we have previously observed designers usually choosing the design that is most similar to a previous design iteration when presented with several designs that are equally good.

Table 4.2: Results of FFDF-ID Analysis after 1 round of iteration shown in Probability of Survival on Demand (PSD)

<table>
<thead>
<tr>
<th>Configuration (# of functions of type)</th>
<th>Probability of Survival on Demand (PSD), For Failure Directed to:</th>
<th>Undirected Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accumulate Energy (Solar Cell)</td>
<td>Store Energy (Battery)</td>
<td>Accumulate Energy (Solar Cell)</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>2.184E-01</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>2.184E-01</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>7.216E-02</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>1.785E-01</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>2.183E-01</td>
</tr>
</tbody>
</table>

During the second iteration, ten accumulate energy and four store energy functions are used as the base configuration. There are five total viable variant functional models that are generated in the second pass through Step 10 of FFDF-ID that are then analyzed while varying the number of store data functions and record visual functions. The viable options include 1 store data and 5 record visual, 2 store data and 4 record visual, 3 store data and 3 record visual, 4 store data and 2 record visual, and 5 store data and 1 record visual function. The lowest survivability is for the 4 store data and 2 record visual with 2.01E-3 PSD, performing worse than the original configuration. The highest survivability is for the configuration with 1 store data and 5 record visual, having a survivability of 0.7828 PSD. However, this is only marginally better than the configuration where there were 5 store data and only 1 record visual, which had a survivability of 0.7819 PSD.

The final selected configuration for the rover should therefore be 10 accumulate energy, 4 store energy, 1 store data, and 5 record visual with failure directed towards the accumulate energy functions representing solar cells. This configuration is found to have a survivability that is 0.7828 PSD when facing failure if the failure is directed towards an accumulate energy function or store energy function representing solar cells and batteries respectively. As a result of the heightened probability of system survivability from directed failure to the accumulate energy and store energy functions, it is recommended the FFDF failure functions be placed into the system in order to direct failure to these functions. A graphical representation of this can be seen in Appendix A.3. As the
mission length increases both the control and improved design survival probability diminishes, but due to their different initial hazard rates the disparity in survival between the improved design and the control design increases dramatically. Further discussion of this behavior can be found below in section 5.2.

Table 4.3: Results of FFDF-ID Analysis after 2 rounds of iteration shown in Probability of Survival on Demand (PSD)

<table>
<thead>
<tr>
<th>Configuration (# of functions of type)</th>
<th>Probability of Survival on Demand (PSD), For Failure Directed to:</th>
<th>Undirected Failure</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Store Data (Memory)</td>
<td>Record Visual (Camera)</td>
</tr>
<tr>
<td>1</td>
<td>7.828E-01</td>
<td>7.828E-01</td>
</tr>
<tr>
<td>2</td>
<td>5.772E-01</td>
<td>5.772E-01</td>
</tr>
<tr>
<td>3</td>
<td>2.184E-01</td>
<td>2.184E-01</td>
</tr>
<tr>
<td>4</td>
<td>2.015E-03</td>
<td>2.015E-03</td>
</tr>
<tr>
<td>5</td>
<td>7.819E-01</td>
<td>7.819E-01</td>
</tr>
</tbody>
</table>

4.6 Discussion

FFDF-ID is shown to be an effective method for improving the survivability of a system. The case study and results (above) shows that the FFDF-ID method is able to inform design decisions and lead to a calculated improved survivability of the system.

While we limit the number of functional model configurations per iteration round and the number of iteration rounds in our case study for ease of understanding, a greater number of functional model configurations can be generated during each iteration round and more iterations can be performed. Additionally, more open-ended design problems can be considered than those used in the case study. More iterations allow for further improvement and refinement of designs, and potentially a higher system survivability. However, more iterations and more functional model configurations come at the cost of computational resources. A balance must be struck by the practitioner. We recommend establishing a cutoff threshold for when to terminate iterations of 1% change between the previous iteration’s best design and the current iteration’s best design.

Several other interesting behaviors are observed in the results. The first notable phenomenon is that in the second phase of the case study, a clear minimum probability of survival emerges in the tradeoff between the store data functions and record visual functions. This phenomenon is observed in all directed and undirected failure cases measured. Upon further investigation, it appears that this configuration has both a very high probability of failing under kNCF conditions and a very high probability of causing failure under ICF conditions. This
illustrate the value of the FFDF-ID method by showing that design decisions potential unpredictable results with major consequences on system survivability.

One application of FFDF-ID that may be able to further increase the survivability of a mission is to use FFDF-ID in the design of individuals in a swarm of robots designed to explore an unknown space [75,76]. In this case, a relatively small increase in individual survivability can lead to a greater survivability of the swarm in its attempt to complete a mission. For example, comparing a swarm 100 robots with the survivability of the initial control design in the case study after one year to a swarm of 100 robots with the final case study design after one year, the probability that at least half of the control swarm will survive is 0.898% and the probability that half the final design swarm will survive is 75.49%. At this scale, FFDF-ID may be very effective at influencing total mission success.

Another potential application of FFDF-ID is in the analysis of all of the systems involved in a space mission through viewing these systems as a single system-of-systems. Through using FFDF-ID, the survivability of each individual system can be increased, thus leading to a potential large increase in overall mission survivability. As in the case of a swarm of robots, a small increase to the survivability of each individual system in a system-of-systems may have a large cumulative effect on the total survivability of the mission.

The FFDF-ID method can have positive impacts in many system design efforts where system failure or downtime is highly undesirable including space exploration, transportation, and power generation. For example, a power generation system design approached with FFDF-ID may lead to a design that is more reliable and has higher system up-time which is important for power generation contracts and grid load balancing. FFDF-ID further can also allow for rapid automated design and analysis of an entire power grid including functions for power generation, distribution, and regulation with a focus on continuous and uninterrupted power.

4.7 Conclusion and Future Work

The FFDF-ID method is an effective method for improving system survivability at the early conceptual phase of system design. FFDF-ID presents a method that uses readily available computational resources in order to create and analyze incredibly large and complex functional models for failure, thus allowing for more complex systems to be designed, analyzed, and
controlled. This is performed through the completion of three phases. The first phase is the
generation of a functional model to represent a system of interest. Phase consists of actions that
build on existing functional modelling techniques, and proposes a convention for the development
of machine interpretable functional models for analysis. The second phase consists of iterative
analysis of the functional model using FFDF methods, while system design is iterated with the
objective of increase probability of system survival when facing directed and undirected failure.
The final phase consists of interpreting the results of the analysis.

4.7.1 Future work

Future development of FFDF-ID will involve improvement of the analysis software for
efficiency and effectiveness. Specific improvements include the development of a user-friendly
Graphical User Interface (GUI) for the creation of functional models. In addition to making FFDF-
ID more approachable to new users, the GUI will allow for the generation of a wide variety of
functional models through crowd sourcing. The shared models may then provide the basis for the
development and analysis of increasingly complex systems through FFDF-ID and other related
methods.

In addition to databases to contain user-generated functional models, data bases of common
components and their functional models in a form that is structured for use in FFDF should be
generated. This would allow for both the more rapid generation of functional models constructed
of these components, as well as create a searchable database of parts and systems that may be
substitutable for each other in a system. For example, a gas generator and a solar power system
may both come up in a search of the database for components that can provide a desired level of
power. In addition to benefits to human designers, a functional model database of components
could be used to enable more sophisticated automated design systems.

Additionally, work should be performed on the implementation of more streamlined
methods for FFDF-ID into other risk analysis techniques such as Active Mission Success
Estimation (AMSE) [51] or autonomous system decision making. This may enable the use of
functional models generated and analyzed through FFDF-ID to be used in the analysis of more
complex missions such as space mission design or to provide the basis for the generation of
autonomous system behavior and decisions.
Acknowledgements

This research was partially supported by United States Nuclear Regulatory Commission Grant Number NRC-HQ-84-14-G-0047. Any opinions or findings of this work are the responsibility of the authors, and do not necessarily reflect the views of the sponsors or collaborators. The authors wish to acknowledge the work of the undergraduate research assistants in the Van Bossuyt lab and specifically wish to thank the following students for their contributions: Alexis Humann, David Hodge, Zachary Mimlitz, and Robin Coleman. The authors wish to thank LeVar Burton, Fred Rogers, Gene Roddenberry, Carl Sagan, and their individual middle school and high school science and technical arts teachers who inspired them to pursue careers in the sciences and engineering, and instilled in them a sense of purpose and compassion.
CHAPTER 5
ACTIVE MISSION SUCCESS ESTIMATION THROUGH PHM-INFORMED PROBABILISTIC MODELLING FOR A CREWED MARTIAN MISSION

A paper to be submitted to Journal of Engineering Design
Ada Rhodes Short, Douglas L. Van Bossuyt

Abstract

Through the application of Prognostics and Health Management (PHM) models and data, the Active Mission Success Estimation (AMSE) introduced in this paper can be performed during a rapidly developing unanticipated failure scenario to support decision-making. AMSE allows for system operators to make informed management and control decisions by performing analysis on a system-of-systems of PHM-informed functional models that requires low time and computational cost. Existing methods for analysis of mission success such as such as Probabilistic Risk Assessment (PRA) or Worst Case Analysis (WCA) have been applied in the analysis and planning of space missions since the mid-20th century. While these methods are effective in analyzing anticipated failure scenarios, they are built on computational models, logical structures, and statistical models that often are difficult and time intensive to modify, and are computationally inefficient leading to very long calculation times and making their ability to respond to unanticipated or rapidly developing scenarios limited. A case study of a generalized crewed Martian surface station mission is presented to demonstrate AMSE. A crew of four astronauts must perform activities to achieve scientific objectives while surviving for 1070 Martian Sols before returning to Earth. A second crew arrives at the same site to add to the settlement midway through the mission. AMSE uses PHM-informed functional models to represent all of the major environments, infrastructure, equipment, consumables, and critical systems of interest (astronauts in the case study presented) in a nested system-of-systems framework that is capable of providing rapidly reconfigurable and calculable analysis. This allows for the AMSE to be used to make informed mission control decisions when facing rapidly developing or unanticipated scenarios. Additionally, AMSE provides a framework for the inclusion of human systems into PHM-based
analysis through a system-of-systems approach. Application of AMSE is expected to produce PHM-informed decision-making benefits in a variety of situations where humans and machines work together toward mission goals in uncertain and unpredictable conditions.

5.1 Introduction

The development of Prognostics and Health Management (PHM) and risk analysis have been deeply linked to space exploration since the formalization of risk analysis methods following the Second World War. Both the era of space exploration and risk analysis of complex systems spawned from the technological progress of the Second World War and the advent of modern rocketry in the early twentieth century [77]. The space race between the United States of America and the Soviet Union spurred the development of tools such as Probabilistic Risk Assessment (PRA) [16] with the aim to closely examine complex system risk probabilistically and quantitatively. At the same time, PHM began to emerge. As increasing complex systems were developed for space flight and exploration, it became imperative that engineers and operators have the ability to accurately and actively monitor system health and performance. Sensors were developed that could monitor every aspect of system operation, including phenomena that would otherwise have been imperceptible. Taking data from these sensors, models of system operation and health could be constructed, laying the groundwork for modern PHM. In recent years, there has been an increased interest in understanding risk and health of systems during the early phase of design of complex systems [11,28,29,78]. However, a gap persists in the development of real-time risk-informed decision support tools for active and ongoing missions. Contemporary mission analysis and risk modeling methodologies require lengthy and extensive adjustment of PHM models and reanalysis when faced with unforeseen events. The subsequent delay of critical risk information necessary for decisions can lead to rapid development of complex and dangerous scenarios.

Through adoption of a modular PHM-informed object-oriented approach to mission modelling, health monitoring, and analysis; and active recalculation of risk of mission failure as the mission progress, a more accurate estimation of the probability of mission success can be developed and mission-critical decisions with many possible options can be analyzed to help inform mission control decision to increase the probability of total mission success. This paper presents the Active Mission Success Estimation (AMSE) method that provides timely risk
information to inform mission decisions being made in crisis situations in rapidly evolving situations.

The performance of AMSE necessitates that all mission-critical components be modelled thoroughly using PHM techniques and the models be developed for modularity to enable the rapid rearrangement of the model elements to evaluate available decision outcomes and estimate each outcome’s mission success probability. In order to effectively represent a mission framework, a functional modelling method is presented where environments of interest and relevance can nest within each other and contain the systems of interest. This system-of-systems approach to modelling is used to determine what environmental hazards are present and can cause damage to the system of interest. Mission tasks that are to be performed are modelled and analyzed through the configuration of tasks to represent internal and external system risks, as well as mitigating factors such as the effects of the nested functional modeling environment representing multiple protection barriers to external and internal initiating events. The AMSE method presented in this paper is demonstrated on a case study of a crewed multiyear scientific mission on the surface of Mars for the establishment of a permanent scientific base. In the case study, the eight astronauts constitute the systems of interest and their safety and survival is considered the metric for mission success.

5.1.1 Specific Contributions

This paper presents the AMSE method for the real-time estimation of risk during a space mission through the utilization of PHM techniques and functional modelling. The AMSE method provides decision-makers with up-to-date risk information at critical mission decision points. The AMSE method uses a form of nested functional modelling to analyze the influence of various layers of environmental protection such as space suits, vehicles, or structures. These protective layers can either provide protection to the subject of interest directly, protect mission-critical systems outside of the subject of interest, or protect each other through layering systems in a nested structure. The AMSE functional modelling technique takes a dynamic system-of-systems approach to provide a comprehensive picture of the interactions between various mission components. AMSE provides a rapid and active estimation of current mission success, as well as projections of probable total mission success based upon potential decisions. Through active analysis of the probability of mission success at decision points, the probability of total mission
success can be optimized allowing for greater mission safety and potentially greater scientific yield. Additionally, the object-oriented modular nature of the AMSE method enables fast adaption to unexpected mission scenarios. Though AMSE was developed for application in risk analysis of space mission operations, AMSE can be easily adapted for use with any complex system and has potential applications for autonomous decision making.

5.2 Background

AMSE builds on the topics of decision theory, functional modeling, risk analysis, and PHM. Existing mission success estimation methods rely on methods such as Worst Case Analysis (WCA) [79,80] or Probabilistic Risk Assessment (PRA) [34,81]. WCA, PRA, and other related methodologies are very adept at analyzing potential foreseeable failure scenarios but suffer in their ability to perform in situations where rapid reconfiguration of the model is necessary to align with rapidly developing situations, such as those faced by in a space mission disaster.

5.2.1 Functional Modelling

Functional modelling encompasses a variety of methods used to represent and model the functionality of a system. Functional models often include many sub-functions representing work performed in the system as flows that represent the passage of materials, information, and energy between functions and sub-functions. In addition to flows internal to the system, export flows and import flows enter and exit the system boundary. A popular way to represent a functional model is through Flow Block Diagrams, also often interchangeably referred to as Functional Flow Diagrams (FFD) [55,62]. FFDs are very useful for modelling systems with direct linear flows passing between a variety of functions and clear system inputs and outputs can be defined. One issue with many existing methodologies for functional modelling is that they are difficult to apply to systems that are less linear, resulting in tangled networks of functions and flows that are difficult or impractical to analyze or must be simplified to the point where they provide an inaccurate representation of the system and its associated dynamics.

The Functional Basis for Engineering Design (FBED) [17,60,82,83], provides concise definitions of functions and flows that describe all possible engineered systems. Through use of FBED functional models can be constructed of complex systems a common taxonomy of functions and flows. The process of developing an FBED model is: 1) Generate a Black Box model. This takes the highest level possible view of the system and only considers flows into and out of the
model. 2) Create function chains for each input flow and order them with respect to time. This step consists of following a flow from its entrance into the system, through all sub-systems that interact with the flow, and finally exiting the system. All systems that interact with the flow should then be placed into chronological order from perspective of the flow. 3) Aggregate function chains into a functional model. In Step 3, the final step of FBED, the functional chains are combined in order to determine the underlying functional structure of the system. FBED is utilized in this paper due to the advanced development of failure analysis methods that are built upon FBED [18,22,24,31,84].

5.2.2 Space Mission Risk Assessment

Risk assessment for space missions can take on many different forms, each with distinct advantages and disadvantages, however they generally tend to build upon a foundation of probability modelling. Many modelling techniques attempt to represent trends of physical failure through the application various failure distributions. One common method is the use of a hazard rate, \( \lambda \), which describes the expected number of failures over a period of time. The hazard rate can be used in a failure distribution such as an exponential distribution, Eq. 5.1, to calculate the probability of survival of a system or sub-system at a given time [64].

\[
S(t) = e^{-\lambda t}
\]  
(5.1)

The expected survival rate can then be subtracted from 1, Eq. 5.2, in order to find the failure rate or the probability that a system will be have survived after time, \( t \),

\[
F(t) = 1 - S(t) = 1 - e^{-\lambda t}
\]  
(5.2)

The failure rate or a related metric appears in a wide variety of risk assessment methods, but many additional and more complex techniques exist for evaluation of risk of failure to a system. One such method for evaluating risk of failure is Failure Flow Identification and Propagation (FFIP) [22,31]. FFIP uses a functional modelling approach based in a function block diagram structure [17]. FFIP can be enhanced in order to enable mission control, navigation, and autonomous decision making through the application of Failure Flow Decision Functions (FFDF) [2]. FFDF is a tool that determines an optimal decision when faced with problems of controlling or designing a system in order to maximize system survivability. Space mission risk assessment
can also be applied to control of autonomous systems in order to maximize mission success while minimizing human work hours [5,6,16,32–34].

While many of the existing methods are generally fairly robust, they suffer from lengthy setup and analysis processes. The lengthy and computationally resource-intensive setup of existing methodologies makes active assessment of dynamic situations infeasible when relying on established methods.

**5.2.3 Prognostics and Health Management**

Prognostics and Health Management (PHM) is a suite of analytical tools and methods used to predict and prevent failures in mechatronic systems [25]. There are diverse approaches to PHM that are mostly tuned to specific applications or industries [19,85]. A common PHM case study for development of models is battery health [86]. Much research has been conducted on the important issues of battery capacity depletion [87], optimization of battery life [88], generation of battery health data [37], and application of battery PHM analysis [91]. While battery health is a common case study, partially due to the large quantity of available data [89] and partially a result of general acceptance within the field, the methods and techniques are generalizable to a wide variety of systems and applications such as electrical actuators [92], transmissions and gearboxes [93], and other components and systems [94].

PHM analysis can be used to choose an option with the optimum level of risk through Prognostic-enabled Decision Making (PDM) [49,95,96]. PDM is a valuable method in health management of complex systems because it allows a succinct modelling of potential damage caused by the failure of a subsystem or individual part. Some PHM techniques model not only the mechatronic system itself, but also of the physical interactions it encounters such as: mobility and environmental interface, control systems, structural actions, and hazards [85,97]. In this paper, we extend PHM methods to include the consideration of humans as additional subsystems which to our knowledge has not been done before.

**5.3 Methodology**

The AMSE method presented here is based on a system-of-systems approach to space mission risk assessment that allows for the active estimation of mission success during an ongoing mission. By using techniques derived from functional modeling of systems, FFIP, and related methods in conjunction with concepts taken from decision theory and PHM, AMSE is capable of
providing useful insights when making mission control decisions by rapidly analyzing potential options when confronted with unanticipated and previously unanalyzed scenarios. This section presents the AMSE method and provides instruction on how to develop an AMSE model for a mission. First, two pre-steps are presented, then three primary phases (modelling, analysis, and interpretation) are shown.

**Pre Steps**

Two pre-steps must be completed prior to the implementation of the AMSE method. First, a definition and quantification of mission success must be identified for the system of interest. Then functional models must be developed of the systems within the mission.

**Pre-Step 1: Mission Success Definition**

In order to glean insight from AMSE, a definitive definition of mission success must first be defined and a quantifiable method for evaluating success must also be established. In many cases, mission success can be defined as a primary system (or systems) of interest surviving the length of the mission. One example of a system of interest surviving the length of a mission is a planetary exploration rover remaining functional for the entire length of the planned mission. In order to determine the probability of survival of a primary system of interest and the related probability of mission success, a survival rate must be calculated. A survival rate, \( S(t) \), tends to take the form of a Cumulative Distribution Function (CDF) representing the probability that the system of interest will not have experienced a failure by time, \( t \). One common form for a survival rate is the exponential survival rate which is found by subtracting the exponential failure rate, \( F(t) \), from 1 as shown in Eq. 5.5. The exponential failure rate is found by taking the integral of the Probability Density Function (PDF) form of the exponential failure rate, \( f(\tau) \), which determines the probability that a failure will occur at the instant, \( \tau \), given a hazard rate, \( \lambda \), which is the number of expected system failures over time. Equations 5.3, 5.4, and 5.5 define \( f(\tau) \), \( F(t) \), and \( S(t) \) respectively. These and other forms of failure distributions, such as system specific PHM models, are an integral part of the AMSE methodology and necessary for the development of failure models.

\[
f(\tau) = \lambda e^{-\lambda \tau} \quad (5.3)
\]

\[
F(t) = 1 - e^{-\lambda t} = \int_0^t \lambda e^{-\lambda \tau} \, d\tau 
\]

\[
S(t) = \int_0^t \lambda e^{-\lambda \tau} \, d\tau 
\]

63
\[ S(t) = e^{-\lambda t} = 1 - (1 - e^{-\lambda t}) \tag{5.5} \]

Pre-Step Two: Model Development

The AMSE method requires a series of functional models to be developed to represent every major system involved in the mission and their individual PHM characteristics. We suggest using the FBED method of functional modeling and do so for the rest of this paper although other methods of system modeling that represent energy, material, and data flows can also be used. PHM information that can be collected from systems in real-time must be identified in this step and tied to the functions whose health the PHM information is tied to. This information is encoded into the mathematical models developed below. The functional models are then put into a system-of-systems framework in the next phase of AMSE so that their interactions with each other may be analyzed.

Phase 1: Modelling

In Phase 1 of the AMSE method, seven distinct steps are performed to develop the AMSE model. Figure 5.1 graphically shows the seven steps. The steps are detailed below.

Step 1: Create a nested functional model of the mission.

The first step consists of modeling all major mission systems (previously modeled in the pre-steps above) within a system-of-systems framework using the PHM-informed functional models developed in Pre-Step 2. Modeling the mission systems within a system-of-systems framework is performed by first modelling each individual system using traditional FBED methods (Pre-Step 2), before placing the individual systems into a nested system-of-systems structure. An example functional model of a Surface Exploration Vehicle (SEV) can be seen in Figure 5.2. A graphical representation of the AMSE system-of-systems structure for a Mars crewed surface exploration mission can be seen in Figure 5.3. In Figure 5.3, the outermost “system” is the space environment in the solar system that contains the Sun, Earth, Mars, and a communications satellite. Within the Earth “system,” mission control is found. The SEV, the Martian Surface Habitat, and the EVA suit are located within the “Mars” system. Within the EVA suit, the astronaut is found. Thus the astronaut (the system of interest in the case study presented in the next section) is inside three larger systems. Under this method, flows can pass between systems while crossing the boundaries of environmental or protective systems such as a SEV, space suit, or Martian surface habitat module. This allows for the entire system-of-systems to be modelled
and represent environmental hazards as well as how various levels of protection work to prevent and mitigate system failure. Additionally, the effects of the current health of each layer of protection on the system of interest can be determined through application of PHM models and information (identified in pre step 2) for each individual system.

Figure 5.1: Phase 1, Modelling, Process
Figure 5.2: Functional Model of an SEV

Figure 5.3: System-of-systems Functional Model
Step 2: Define Critical System(s) of Interest and Critical Flows

In the case of a functional model of a single system, critical functions and flows are often defined as elements of the functional model that must perform their intended functions or flows nominally for the system to not be in a failure state [65]. In the context of a system-of-systems representing a mission framework, the idea of critical functions and flows is extended from the functional level to the system level, and a critical system of interest is defined. A critical system (or systems) of interest is a system that must be functioning in order for the mission to be considered not failed. For example, in the case of a rover mission, the critical system of interest is the rover, and for the case of a crewed space mission, each member of the crew is considered a critical system of interest¹. Step 2 concludes once the critical system(s) has been identified and defined.

Step 3: Develop Mathematical Models to Represent Graphical Functional Models, Their Health, Failure Distributions, and How Failures Relate to Each Other

The third step of the AMSE method consists of developing a mathematical model to represent the graphical functional model and PHM information developed in the second Pre Step. This mathematic model serves as the computational basis of analysis of the system. Building on previous work on failure analysis and PHM in functional models, the logic by which failure propagates can be described and analyzed [1,2].

In the AMSE method, it is important to assign failure distributions to systems and accurately represent how failure is passed between systems [98]. PHM-based failure distributions must be selected that are dependent on the flows passed into and out of the system, and often are dependent on the time over which the system is utilized (though not exclusively, and could be dependent on resources such as the flow of cooling fluid at appropriate levels or available energy). Several common forms of failure distributions to use involve the Weibull distribution, normal distribution, and the exponential distribution [98]. Additionally, for many systems, more complex

¹ Except in the case of a Star Trek: The Original Series (ST:TOS) “Red Shirt” scenario where expendable crew members are present. Note that the authors of this paper do not condone Red Shirt mission designs, especially without the consent of said Red Shirt-designated crew.
prognostic health models have been developed and can be implemented into the math of the system models [99–102].

Once the individual systems have been analyzed in order to determine how failure will propagate [1,21,103], the system of systems assembled in Step 1 can be modelled. The system of systems model is constructed in the same manner as a single functional model, but with systems in the place of sub-systems. The end product is a mathematic representation of a PHM-informed functional model that can track the passage of flows between all mission systems and actively reported an estimated system health.

*Step 4: Define a Mission Plan*

A mission plan is used in AMSE to develop future scenarios for automatic mission success probability calculation. The mission plan includes the planned operations and objectives to be completed over the course of a mission. We suggest that the mission plan start loosely with only primary mission objectives and milestones defined at first, and then the secondary objectives and operations that must be completed in order to facilitate the performance of objectives can be developed. For use with AMSE, the mission plan is then broken down further into actionable items that can be completed by systems in the mission. These actionable items are referred to as “tasks” for the rest of this paper. Examples of tasks for a rover include driving a specific distance, performing a scientific operation, or performing communication with Earth. For the case of a crewed space mission, tasks may include EVAs, performance of experiments, or health-related tasks such as eating and sleeping.

*Step 5: Develop Task Modules*

Task modules are important to develop for the AMSE method because AMSE uses tasks to automatically plan how mission objectives can be completed when analyzing potential decision choices. Tasks modules include the duration that a task is to be performed, all systems and resources used during the task, and any fatiguing or consumption of systems affecting health of systems that may occur during completion of the task. This information will be necessary for analyzing the mission in Phase 2 of AMSE. Table 5.1 lists several typical mission tasks, and associated resource and system health cost parameters.
Table 5.1: Typical Mission Tasks and Associated Costs

<table>
<thead>
<tr>
<th>Task</th>
<th>Duration (sec)</th>
<th>Systems Used</th>
<th>System Health Factors</th>
<th>Resources Used</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sleeping</td>
<td>30960</td>
<td>Habitat Module</td>
<td>Time Inhabited 30960 (sec)</td>
<td>Calories Burned</td>
<td>~8.5 (kcal/kg)*Astronaut Weight (kg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Physical Intensity 0.5/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Eat Food</td>
<td>7200</td>
<td>Habitat Module</td>
<td>Time Inhabited 7200 (sec)</td>
<td>Calories Burned</td>
<td>~2.8 (kcal/kg)*Astronaut Weight (kg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Physical Intensity 1.0/10</td>
<td>Food Eaten</td>
<td>3025 (kcal) Gained</td>
</tr>
<tr>
<td>Exercise</td>
<td>7200</td>
<td>Habitat Module</td>
<td>Time Inhabited 7200 (sec)</td>
<td>Calories Burned</td>
<td>~17.4 (kcal/kg)*Astronaut Weight (kg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Physical Intensity 9.5/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintain Farm</td>
<td>3600</td>
<td>Farm Module</td>
<td>Time Inhabited 3600 (sec)</td>
<td>Calories Burned</td>
<td>~4.4 (kcal/kg)*Astronaut Weight (kg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Physical Intensity 4.5/10</td>
<td>Water Used</td>
<td>~20 (L/m² of Crops Being Grown)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Food Produced ~8.4 (kg/day) At full production</td>
</tr>
<tr>
<td>EVA</td>
<td>28800</td>
<td>Air Lock</td>
<td>Uses 2</td>
<td>Calories Burned</td>
<td>~25 (kcal/kg)*Astronaut Weight (kg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EMU</td>
<td>Time Inhabited 28800 (sec)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Physical Intensity 3.0/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>SEV</td>
<td>Time Inhabited 3600 (sec)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Physical Intensity 1.7/10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IVA</td>
<td>10800</td>
<td>Habitat Module</td>
<td>Time Inhabited 10800 (sec)</td>
<td>Calories Burned</td>
<td>~5 (kcal/kg)*Astronaut Weight (kg)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Physical Intensity 1.3/10</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Step 6: Organize Tasks into a Task Plan**

Using the task modules generated in Step 5, the next step is to organize the task modules into a task plan that defines typical operations or schedules that are to be followed within the mission plan. For example, a task plan can represent all of the tasks to be completed on a particular type of day, such as a day that an EVA is to be performed by a crew member. Additionally, a typical week can be assembled from task plans for days and made into a larger meta-task plan. The bundling of task modules into task plans allows for more rapid reconfiguration of the system-of-systems model for analysis by AMSE by allowing the mission controller or astronaut performing the analysis to quickly assemble a typical period of time to include into the analysis.

**Step 7: Arrange Task Plans to Align with the Mission Plan**

The general mission plan defined in Step 4 is now filled in with task plans developed in Step 6. This enables the analysis of the mission using AMSE by providing a time-discretized list of all of the actions and systems that are to be used for completion of the mission as a whole. Figure 5.4 shows how task modules are assembled into task plans and then arranged to align with the mission plan.
While each of the seven steps of Phase 1 must be completed prior to using AMSE, and the initial modelling can involve a large time investment, once many of these steps have been performed they do not have to be performed again. If the model needs to be reconfigured in order to account for an unforeseen circumstance or to iterate on the mission design (in the case of using AMSE for mission design rather than mission operations), adjustment of the models developed in Step 3 or reconfiguration of the Task Plans in Step 6 can account for the majority of changes that may need to occur to the mission plan and its constituent parts. Due to the ease of configurability enabled by initial up-front investment of time and resources in model building, AMSE models are able to be reconfigured rapidly to adjust to unforeseen circumstance or examine a variety of options in order to inform a mission control decision.

**Figure 5.4: Organization Structure of Tasks**
Phase 2: AMSE Analysis

As with Phase 1 of AMSE, the second phase, analysis, requires the investment of time and resources to generate the mission models for analysis. Unlike Phase 1, Phase 2 only must setup once and will be run whenever evaluation of a new mission model is desired. The majority of the math necessary for Phase 2 was already developed from Step 3 of Phase 1 where the mathematical
PHM representation of the mission was developed. The performance of Phase 2 takes the form of execution of an algorithm consisting of eight individual steps. The eight steps that compose the Phase 2 algorithm are detailed below. A flowchart of Phase 2 algorithm can be seen in Figure 5.5.

**Step 1: Step Through Mission Plan**

Step through the mission plan generated in Phase 1. Starting with the earliest task that has not yet been analyzed select each task and then perform Steps 2 through 5 on them. This is necessary to analyze how the success rate of the mission develops over time.

**Step 2: Calculate Resource Cost of Task and PHM Effects**

Calculate resource cost of task completion and PHM effects on systems used. Any resources consumed or systems fatigued by the completion of the task must be accounted for. One implementation of this is a resource matrix that contains how much of each resource is available and subtract from the matrix as resources are consumed. A similar approach can be utilized for the tracking of system health from mechanical wear, environmental conditions, or energy usage.

As an example of Step 2 of the algorithm, the model for kilocalories used by an astronaut during the performance of a task is displayed in Eq. 5.6, where $k$ represents kilocalories used, $p$ represents physical exertion required to perform a task on a scale of 0 to 10, where sleep is a 0.5 and vigorous exercise is a 9.5, $d$ represents the duration of the task in hours, and $w$ represents the astronauts current weight in kilograms.

\[
k = (p \cdot 0.8556 + 0.5622) \cdot w \cdot d \tag{5.6}
\]

**Step 3: Calculate Hazard Rates Presented to Critical System of Interest**

Calculate the risk presented to the critical systems of interest. Utilizing the mathematical system model with PHM information developed in Phase 1, calculate what the risk of system failure is for completion of the task. We recommend calculating the risk in the form of a hazard rate, $\lambda(\tau)$, representing the number of system failures expected at the instant $\tau$.

An example of Step 3 is the equation used to calculate the hazard rate for completion of a task presented to an astronaut, this shows how generic PHM models can be adapted from existing data when specific models have not yet been developed. This is determined by calculating the critical system of interest’s (such as robots or humans) expected failures per day for all potential causes of failure combined. The considered potential causes of critical system failures for an
astronaut are radiation poisoning, hypothermia or hyperthermia, starvation, lack of sleep, or injury from physical harm (such as from severe impact). While hazard rates for mechanical systems can be simply calculated by determining the number of expected failures per hour, actuarial information for the loss of a human tends to be in terms of expected time suffering from a condition before loss [104]. However, by inverting this data, an expected failure per day value can be determined. In order to calculate the expected hours a system is exposed to a hazard before loss of life, distributions are fit to prognostic information using metrics calculated within the model. For the case of starvation, a cumulative distribution function for a normal distribution, \( N(\mu, \sigma^2, X) \), of the Body Mass Index (BMI) is used with a mean of 14 kg/m\(^2\). This distribution is chosen because it allows for a rapid transition between a typical life expectancy, \( L \), and an incredibly short life expectancy, with very little transitional space between the two. BMI is calculated using the current weight of the astronaut in kg, \( w \), divided by the astronaut’s height in meters, \( h \), squared. This best represents available data. The hazard rate for starvation as a function of BMI is shown in Equation 5.7. A generalized form is shown as Equation 5.8 in Formulation 1 below.

\[
\lambda_{starve} = N(14,1, BMI)^{-1} = 2 \left[ 1 + \text{erf} \left( \frac{w}{h^2} - 14 \right) \right]^{-1}
\] (5.7)

One consequence of modelling hazard rates this way is that the distributions fitting process could be accurate in one region at the cost of accuracy in another region. In order to make the models more useful, they are fit so that the distributions are more accurate when in areas of interest such as near regions were failure is likely to occur, but may be unrealistically optimistic regions where no risk was presented. One way to overcome this limitation is through the use of less generalized distributions collected from case-specific experimental data or through the use of pre-existing PHM models whenever available.

**Step 4: Record Hazard Rates**

Record hazard rates and time of occurrence. A matrix containing hazard rates for the systems of interest and the time at which the hazard rate was reached should be generated. This will be necessary for the calculate of a total mission failure and success rate in later steps. The matrix values for the first three Sols spent on Mars for one of the astronauts in the case study presented in this paper is reported in Table 5.2.
Step 5: Repeat Until Complete

If tasks still exist in the mission plan that have not yet been analyzed, return to Step 1 of Phase 2. If all tasks in the mission plan have been completed, then continue on to Step 6 of Phase 2.

### Table 5.2: Hazard Rates for Failure Type [Failure/hour]

<table>
<thead>
<tr>
<th>Sol</th>
<th>Hour</th>
<th>Radiation</th>
<th>Temperature</th>
<th>Starvation</th>
<th>Exhaustion</th>
<th>Injury</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>8.6</td>
<td>1.53E-06</td>
<td>5.09E-80</td>
<td>7.48E-12</td>
<td>9.87E-10</td>
<td>1.00E-09</td>
</tr>
<tr>
<td>0</td>
<td>9.6</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>1.25E-08</td>
<td>1.37E-07</td>
<td>8.00E-06</td>
</tr>
<tr>
<td>0</td>
<td>13.6</td>
<td>1.53E-06</td>
<td>1.32E-37</td>
<td>1.56E-08</td>
<td>2.33E-07</td>
<td>7.20E-06</td>
</tr>
<tr>
<td>0</td>
<td>15.6</td>
<td>1.53E-06</td>
<td>3.63E-19</td>
<td>1.64E-08</td>
<td>1.74E-06</td>
<td>1.00E-08</td>
</tr>
<tr>
<td>0</td>
<td>19.6</td>
<td>1.53E-06</td>
<td>1.32E-37</td>
<td>2.04E-08</td>
<td>4.50E-06</td>
<td>7.20E-06</td>
</tr>
<tr>
<td>0</td>
<td>24.6</td>
<td>1.53E-06</td>
<td>2.19E-28</td>
<td>1.24E-08</td>
<td>2.67E-05</td>
<td>1.00E-08</td>
</tr>
<tr>
<td>1</td>
<td>33.2</td>
<td>1.53E-06</td>
<td>5.09E-80</td>
<td>8.49E-12</td>
<td>9.87E-10</td>
<td>1.00E-09</td>
</tr>
<tr>
<td>1</td>
<td>36.2</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>7.96E-09</td>
<td>1.37E-07</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>1</td>
<td>37.2</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>8.51E-09</td>
<td>2.33E-07</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>1</td>
<td>38.2</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>9.10E-09</td>
<td>3.91E-07</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>1</td>
<td>39.2</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>9.73E-09</td>
<td>6.49E-07</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>1</td>
<td>40.2</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>1.04E-08</td>
<td>1.07E-06</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>1</td>
<td>41.2</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>1.11E-08</td>
<td>1.74E-06</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>1</td>
<td>42.2</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>1.19E-08</td>
<td>2.81E-06</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>1</td>
<td>43.2</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>1.27E-08</td>
<td>4.50E-06</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>1</td>
<td>47.2</td>
<td>1.53E-06</td>
<td>1.32E-37</td>
<td>1.41E-08</td>
<td>7.12E-06</td>
<td>1.00E-08</td>
</tr>
<tr>
<td>1</td>
<td>49.2</td>
<td>1.53E-06</td>
<td>3.63E-19</td>
<td>1.82E-08</td>
<td>4.07E-05</td>
<td>2.00E-05</td>
</tr>
<tr>
<td>2</td>
<td>57.8</td>
<td>1.53E-06</td>
<td>5.09E-80</td>
<td>1.38E-11</td>
<td>9.87E-10</td>
<td>1.00E-09</td>
</tr>
<tr>
<td>2</td>
<td>60.8</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>1.17E-08</td>
<td>1.37E-07</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>2</td>
<td>61.8</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>1.25E-08</td>
<td>2.33E-07</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>2</td>
<td>62.8</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>1.33E-08</td>
<td>3.91E-07</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>2</td>
<td>63.8</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>1.42E-08</td>
<td>6.49E-07</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>2</td>
<td>64.8</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>1.52E-08</td>
<td>1.07E-06</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>2</td>
<td>65.8</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>1.62E-08</td>
<td>1.74E-06</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>2</td>
<td>66.8</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>1.73E-08</td>
<td>2.81E-06</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>2</td>
<td>67.8</td>
<td>1.53E-06</td>
<td>6.02E-10</td>
<td>1.85E-08</td>
<td>4.50E-06</td>
<td>1.00E-05</td>
</tr>
<tr>
<td>2</td>
<td>71.8</td>
<td>1.53E-06</td>
<td>1.32E-37</td>
<td>2.05E-08</td>
<td>7.12E-06</td>
<td>1.00E-08</td>
</tr>
<tr>
<td>2</td>
<td>73.8</td>
<td>1.53E-06</td>
<td>3.63E-19</td>
<td>2.64E-08</td>
<td>4.07E-05</td>
<td>2.00E-05</td>
</tr>
</tbody>
</table>
Step 6: Calculate Total Mission Hazard Rate

Calculate the total mission hazard rate for remaining mission time, \( \Lambda_a(t) \). Taking the instantaneous hazard rates generated from the functional models and real-time PHM information developed in Steps 2 and 3, calculate the total hazard rate for the remainder of the mission time as a function of time over the entire length of the mission. Like the instantaneous rate, \( \lambda_a(\tau) \), the total mission hazard rate, \( \Lambda_a(t) \), describes the number of expected system failures per unit of time. While this can be found using integration of continuous data, for the purpose of discretized data generated in performance of the AMSE method a weighted average can find the total mission hazard rate. This is found by summing the product of the instantaneous hazard rate for a task and the duration of a task, \( \Delta\tau \), and then dividing by the total mission length, \( T \), minus the current time of the mission as seen in Eq. 5.10 in Formulation 1 below.

Formulation 1: \( \Lambda_a(t) \), Formulation of Hazard Rate

Here we provide the mathematical formulation for \( \Lambda_a(t) \), the hazard total rate presented to a critical system of interest from an environmental or internal hazard over the remaining course of the mission \( \frac{\text{Losses of System}}{\text{mission}} \).

Formulation 1.1: Sets

\( \varepsilon \in E \): Set of all Tasks in a Task Plan

\( \varepsilon \in E_t \): Set of all uncompleted tasks in the task plan after time, \( t \)

\( h \in H \): Set of all hazards faced by the system of interest

\( h \in H_\varepsilon \): Set of all hazards presented to a critical system of interest, \( s \), in the completion of task, \( \varepsilon \)

\( a \in A \): set of all critical systems of interest in a system-of-systems

\( p_{a,h} \in P \): Set of all parameters used to calculate hazard rates in PHM based failure distribution, \( R \), for system, \( a \) [various units]

Formulation 1.2: Parameters

\( T = \text{Total planned mission length [hours]} \)

\( \Delta\tau_\varepsilon = \text{Time elapsed during the completion of a task, \( \varepsilon \) [hours]} \)
Formulation 1.3: Variables

\[ \tau = \text{Instantaneous time in the mission [hour]} \]

\[ t = \text{Time elapsed since mission start [hours]} \]

Formulation 1.4: Calculation

The hazard rate for an individual hazard, \( h \), is found by putting the appropriate parameters into the PHM based failure distribution, \( R \)

\[
\lambda_{ah} = R(p_{hi}) \left[ \frac{\text{Losses of System}}{\text{hour exposed to hazard}} \right]
\]

The total hazard rate presented to a critical system of interest, \( s \), during a task, \( \varepsilon \)

\[
\lambda_a(\varepsilon) = \sum_{h \in H_{\varepsilon}} \lambda_{ah} \left[ \frac{\text{Losses of System}}{\text{hour}} \right]
\]

The combined hazard rate presented to critical system of interest, \( s \), for the remainder of the mission

\[
\Lambda_a(t) = \sum_{\varepsilon \in E_T} \lambda_a(\varepsilon) \cdot \Delta \varepsilon \left[ \frac{\text{Losses of System}}{\text{Mission}} \right]
\]

Step 7: Calculate Probability of Mission Survival Over Time

Calculate the probability of mission survival over time, \( S_a(t) \), for a critical system of interest, \( a \), using the total mission hazard rate as shown in Equation 5.11. In the case of a single critical system of interest, \( S_a(t) \), is equivalent to the total mission probability of success, \( p_{success} \). However, in the case of multiple critical systems of interest, \( p_{success} \) is equivalent to the intersection of the probability of mission survival, \( S_a(t) \), for all systems as shown in Equation 5.12 Formulation 2 below.

Formulation 2: \( (p_{success}) \) Formulation of Probability of Mission Success

Here we provide the mathematical formulation for \( (p_{success}) \), the probability of total mission success \([\text{Successful Missions per Attempt}]\).
Formulation 2.1: Sets
\[ a \in A: \text{set of all critical systems of interest in a system-of-systems} \]

Formulation 2.2: Parameters
\[ T = \text{Total planned mission length [hours]} \]

Formulation 2.3: Variables
\[ t = \text{Time elapsed in the mission so far [hours]} \]

Formulation 2.4: Calculation

The Probability of survival for a single critical of interest, \( s \), is calculated for planned mission time remaining, \( T - t \)

\[ S_a(t) = e^{-\Lambda_a(t) \cdot (T-t)} \left[ \frac{\text{Systems Survive Mission}}{\text{Attempt}} \right] \] (5.11)

The probability of total mission success is calculated for mission time, \( t \)

\[ p_{\text{success}}(t) = \bigcap_{a \in A} S_a(t) \left[ \frac{\text{Successful Missions}}{\text{Attempt}} \right] \] (5.12)

Step 8: Display Results

Finally, results of AMSE analysis is presented in human-readable form to support decision-making. In order to make the results of the AMSE analysis human readable, the instantaneous hazard rate and survival rate for an individual critical system of interest should be plotted as well as the probability of total mission success over time. This provides a quick visual check of how the probability of mission success develops over time, as well as providing insight on any task or period of time that may be adversely affecting the probability of mission success. Additionally, it may be helpful to plot system and hazard specific PHM values that have an influence on mission success in order to determine what degraded system health states may be leading to less than desired mission success probability that need to be addressed directly. Viewing the results of the analysis in this way allows for easier interpretation of the results and troubleshooting of low success probability mission plans and allows prognostics-enabled decisions to be made by human operators that better consider how system health develops over time.
Similarly, to Phase 1, the initial setup of Phase 2 can be time intensive, but after it is set up the first time it is unlikely to require any additional work be performed and it should be applicable to any model generated in Phase 1.

**Phase 3: Interpretation of Results**

Phase 3 of the AMSE method consist of interpreting the results of the analysis from Phase 2. This phase is difficult to break into concise steps as it is less procedural and instead aims to generate mission decision or design insight that is informed by PHM and is model- and mission-specific. However, there is some general advice that can apply to most cases that a practitioner might encounter.

One important metric to observe is the probability of mission success at the beginning of the mission, \( P_{\text{success}}(0) \), or the probability of total mission success over the entire span of the mission from beginning to end. This metric is important because it describes the total probability that a mission will be successful including all tasks, systems, expected environmental conditions, and other PHM-affecting factors over the entire mission plan. Additionally, it should be noted that \( P_{\text{success}}(t) \) at time \( t = 0 \) is the lowest that it will ever be during a nominal mission because it includes all of the risk from all of the tasks that are to be completed.

One way to conceptualize \( P_{\text{success}}(0) \) is as the probability that a speeding driver will be pulled over by the police during a long trip. At the beginning of the drive, there exists the most opportunities for the driver to be pulled over. However, over the course of the trip the number of remaining chances to be pulled over decreases, because there is less of a distance left to traverse, and therefore, less of a chance that the speeding driver will be caught.

Additionally, it should be noted that \( P_{\text{success}}(t) \) approaches 1 as time remaining in the mission approaches 0. It is important to keep this in mind, especially in high risk missions that appear to become more successful near the end of the mission. This line of thought constitutes a fallacy in the way the model is viewed as the higher probability of survival near the end can only be achieved if a low probability of survival is completed near the beginning. Additionally, it is important to understand how a single high risk mission task could drastically lower all of the mission success estimation before the task is completed. For example, if a mission is conducted where all mission tasks have a 100% probability of success, except for one task that has a 10%
chance of success but presents no long term system health effects, the probability of mission success will be only 10% until after the task is completed.

5.4 Case Study

A case study is presented in this section of a hypothetical space mission to establish a permanent research settlement on the Martian surface. The case study presented and models used should not be used in the planning of an actual space mission as AMSE is highly mission-specific. In order to insure that the presented case study is not used for the planning of an actual mission, certain aspects of the mission framework were simplified and some PHM models used were made intentionally unrealistic although still representative. Additionally, this approach allows for the more direct evaluation of the AMSE methodology as a decision support tool while using the case study as a framework for evaluation of AMSE’s effectiveness and responsiveness.

The planned mission consists of two crews consisting of four female astronauts each arriving at the same site 26 months apart. The time horizon of the mission begins with the arrival of the first crew, Crew Alpha, and continues up to their departure after 1070 Martian Sols. This time horizon was selected so that the comparatively high risk activities of accent and descent from orbit would not affect the analysis and the focus can remain on surface operations, and the demonstration of AMSE. The second crew, Crew Beta, is also analyzed with AMSE, but the primary focus of the case study is on Crew Alpha.

5.4.1 Crew Composition

Each crew consists of four female astronauts who are all approximately 170 cm tall and range from 60 to 65 kg. The reason behind sending an all-female crew is that it cuts down on the quantity of food necessary to sustain their health and allows for more shared resources such as commonly sized space suits or Extravehicular Mobility Units (EMUs). This idea has been proposed in the past by a variety of individuals including participants in the NASA Hawaii Space Exploration Analog and Simulation (HI-SEAS) test [105,106].

In order to model human crew survival from a PHM perspective, models of the Martian environment and the necessary conditions for human life are developed. Critical information used in the development of the model is presented in Sections 4.2 through 4.5.
5.4.2 Human Requirements to Live in Space

Humans operating in space environments requires external life support systems to continue living and to be able to perform work tasks. The major requirements for sustained human survival in space include: temperatures between 4-35° C, 0-0.5% atmospheric carbon dioxide by volume, 35-350 kPa ambient pressure, radiation dose below 15 roentgens per year [107], 2 liters of water per day [108], access to 34 essential nutrients [109], and a minimum of approximately 1300 kcal per day [110].

On Mars, threats to maintaining human life include: exposure to radiation, surface storms, and exposure to the very low atmospheric pressures and temperatures. On the Martian surface, ambient pressures averages 0.6% of Earth sea-level pressure, atmospheric composition consists of over 96% carbon dioxide [111], mean surface temperatures are approximately -63°C, and raw surface radiation exposure is upwards of 1000 times greater on the surface of Mars than Earth [112].

5.4.3 Human Exploration of Mars and Site Selection

Current NASA deep space mission planning methodology is heavily reliant on materials acquired at the site through the process of In Situ Resource Utilization (ISRU) [113]. For this reason, NASA has compiled a series of parameters that are ideal for a Mars base site. A decision matrix, compiled by the First Landing Site/Exploration Zone Workshop for Human Missions to the Surface of Mars, lists two primary criteria categories: 1) Scientific Merit and 2) ISRU/Engineering criteria [114]. The engineering criteria considers foundational factors such as water supply and the presence of plant micronutrient minerals that are foundational to a long-term human presence. The optimal ISRU/Engineering selection criteria were used as the primary criteria for site selection.

The principle location risk was deemed to be dust storms. These have, typically, originated in the southern hemisphere during, or around, perihelion, and Martian summer [115]. Dust storms can reduce visibility over the entire planet, making navigation difficult for astronauts caught in a storm during an Extra Vehicular Activities (EVA’s). Additionally, dust can also compromise solar power generation. Evidence for surface lightning has also been observed, which could affect power systems [116]. Dust storms occur at an average rate of 7.1 storms per Martian year [117],
and are generally more intense in the southern hemisphere [118]. Thus, the Northern Hemisphere was is preferable for colonization.

The planned Mars missions utilizes solar power [119]. While average insolation is greater at the poles, it is more consistent at the Martian equator [29]. An average insolation of 200 W/m^2 occurs around the Martian equator. A peri-equatorial site would therefore be best for power and agricultural performance.

Within these criteria, NASA has listed a few potential landing sites for un-manned missions that exhibit fluvial features and possible hydraulic soil infiltrates for ISRU water reclamation. The list includes the Mawrth Vallis and Nili Fossae sites. Martian surface spectroscopy data [30] suggests that the essential micronutrients and minerals vital to the growth of most plants [45] can be found in Martian soil. For this simulation it is assumed that all inorganic plant micronutrients are present at the chosen Martian Sites.

### 5.4.4 Nutrition Requirements

The most important long term life support risk to humans on any deep space mission is nutrition, because food is the greatest one-time consumable by mass after fuel. Lifting mass out of orbit is extremely costly, thus the total supply of food that can be taken into space is limited. Additionally, the biosphere in which most food is grown is arguably one of the most complicated systems yet documented; artificial replication is very prone to catastrophic cascading failure [120]. Therefore, a high risk of starvation exists due to; food production being prone to failure, and food carrying capacity at launch being extremely limited.

The US Food and Drug Administration defines 34 key macro and micro nutrients essential to human survival [121]. In addition to the Daily Recommended Value (DRV), each macro and micro-nutrient has an approximate biological half-life. In order to consolidate this information into a more concise metric, an index of criticality was developed as shown in Equation 5.13.

\[
P_P E E = \frac{P_C}{D_D D_D} B_B C_C L_L - H_H C_C F_F a_a\]  \(5.13\)

This ratio inflates for both high intake requirements and quick biological half-lives, yielding a metric whereby the largest numbers represent the most critical nutrients. Conveniently, this criticality index also indicates which micronutrients are practical to bring from Earth as a
supplements. This index was used to categorize the nutrients that would be more efficient to produce in situ on Mars. Again, high-mass requirements for some consumables, such as carbohydrates, protein, fat, and other macronutrients, restrict the efficiency of supplying such materials from Earth. All macronutrients, namely: carbohydrates, fat, protein, and dietary fiber can only be efficiently produced on site [119]. It was found that the most critical nutrients are, carbohydrates, protein, dietary fiber, and fat.

Crops were selected using two criteria: the aforementioned nutrient criticality index, and growing time. Ultimately, potatoes, soy beans, sweet potatoes, wheat, and peanuts were chosen as the primary crops. Various other crops were considered as well for their rich micro-nutrient production including: cabbage, tomato, bell pepper, spinach, cucumber, kale, garlic, onion, and broccoli. Additionally, it should be noted that several vitamins and minerals are principally animal products and will be assumed to be brought along from launch as dietary supplements. These include cholesterol, vitamin D, Vitamin B12, Vitamin H (biotin), and iodine.

5.4.5 Included Model Systems and Resources

In addition to the models of the astronauts, two Martian surface habitat modules, two SEVs, and twelve total space suits are included (6 space suits brought by Alpha Crew and 6 space suits by Beta Crew). The modelled systems are broken down further into sub-systems such as those for power generation, life support, in-situ resource utilization, or waste management in the case of the habitats. For instance, the PHM model for the habitat examines the quantity and intensity of physical work performed, power consumption, load on the life support systems, time of exposure to the Martian environment, and accumulated fatigue from use of the habitat airlock. Another system for which a PHM model was developed is the SEV, which models the hazard rates of wheel failure, battery loss, mechanical fatigue, and general health effects from exposure to the Martian environment. Equations 5.14 through 5.17 show the distributions used for the hazard rates for tires, power, mechanical fatigue, and environmental damage. Equation 5.18 shows how the combined SEV failure hazard rate is found.

\[
\lambda_{wheel} = 6 \cdot \lambda_{wheel} = \frac{6}{1425} \cdot \left(\frac{t_{driven}}{1425}\right)^4 e^{-\left(t_{driven}/1425\right)^5 \left[\text{expected failure} \right. \text{hour}}} \tag{5.14}
\]

\[
\lambda_{power} = \frac{1}{3600} \cdot \left(\frac{Q_{cycle}}{3600}\right)^5 e^{-\left(Q_{cycle}/3600\right)^6 \left[\text{expected failure} \right. \text{hour}}} \tag{5.15}
\]
\[ \lambda_{mech} = \frac{1}{2425} \cdot \left( \frac{\sum I \cdot t_i}{2425} \right)^3 e^{-\left( \frac{t_{driven}}{2425} \right)^4} \left[ \frac{\text{expected failure}}{\text{hour}} \right] \]  

\[ \lambda_{expo} = \frac{1}{10000} \cdot \left( \frac{t_{mission} - t_{maint}}{10000} \right)^4 e^{-\left( \frac{t_{mission} - t_{maint}}{10000} \right)^5} \left[ \frac{\text{expected failure}}{\text{hour}} \right] \]  

\[ \lambda_{SEV} = \lambda_{wheel} + \lambda_{power} + \lambda_{mech} + \lambda_{expo} \left[ \frac{\text{expected failure}}{\text{hour}} \right] \]  

The SEV allows for greater mission scientific yield through expanding the range of EVAs, but is not necessary to preserving health, so Weibull distributions are fit to desired failure rate characteristics. These distributions can be replaced with more system-specific PHM models in order to increase model accuracy in exchange for minimal computational cost. However, for the purposes of the case study, namely to demonstrate AMSE, the models presented above are sufficient. The hazard rate for the SEV’s wheels, \( \lambda_{\text{wheel}} \), is dependent on the time that the SEV is driven on the Martian surface, \( t_{driven} \), and models six wheels designed to last two whole mission lengths before replacement. The SEV’s battery health, \( \lambda_{\text{power}} \), is dependent on the number of battery change cycles, \( Q_{\text{cycle}} \), with the equivalent cycles of five missions before failure. A larger number of missions before expected failure was used because replacement of the SEV battery would be more time and resource intensive than replacement of the wheels. The SEV’s general mechanical failure rate, \( \lambda_{\text{mech}} \), is dependent on the intensity at which the SEV is driven, \( I \), and the time driven at intensity, \( t_i \), with the equivalent cycles of two mission cycles at expected intensity before failure. The SEV’s failure from exposure to the Martian environment, \( \lambda_{\text{expo}} \), is dependent on the time that has elapsed since the last general maintenance operation, \( t_{\text{mission}} - t_{\text{maint}} \), with the equivalent time between maintenance of 350 Martian sols.

Additionally, a variety of consumable resources are brought along, such as food and the supplies necessary to start a farm in order to generate food and become Earth independent. The crops brought along include soybeans, potatoes, peanuts, wheat, and sweet potatoes. The selection of these crops is informed by previous studies, but new calculations are performed to estimate the volume of each crop to grow including updated nutritional information for crops and metabolic model for caloric intake [119,122]. These crops are chosen for their ability to meet DRV for necessary macro-nutrients, and provide a variety in diet. The crops are grown in a vertical farming unit attached to the Martian habitats. It is assumed that the Martian habitats are deployed before the arrival of the crews and only final verification operations must be performed upon arrival.
5.4.6 Mission Plan

The plan consists of eight stages. The stages are defined as: 1) Alphas arrival and setup 2) Starting Farm Alpha 3) Alpha primary exploration window 4) Preparation for arrival of Beta 5) Start Farm Beta 6) Crew Beta arrival and setup 7) Cooperative scientific window between Alpha and Beta and 8) Preparations for departure of crew Alpha. On a typical day crew members will get 8.6 hours allocated for sleep/hygienic activities, 2 hours for food preparation and eating, 2 hours for exercise, 1 hour for farming, and then the remaining time split between Intra-Vehicular Activities (IVA) and Extra-Vehicular Activities. IVAs refer to any scientific, maintenance, or other task that is performed within the Martian surface habitat module that is not described by another category and EVAs refer to any activities performed in an outside of the habitat while wearing an EMU. This includes tasks that involve the use of the SEVs. EVAs are performed on a rotating nine-sol schedule which can be seen in Table 5.3. On days where an EVA is performed it is typically an 8-hour EVA. The remaining time of the day is dedicated to IVA.

Table 5.3: Crew EVA schedule over a Nine-Sol period

<table>
<thead>
<tr>
<th>Crew Member</th>
<th>Sol 1</th>
<th>Sol 2</th>
<th>Sol 3</th>
<th>Sol 4</th>
<th>Sol 5</th>
<th>Sol 6</th>
<th>Sol 7</th>
<th>Sol 8</th>
<th>Sol 9</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>EVA</td>
<td></td>
<td>EVA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>EVA</td>
<td></td>
<td>EVA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>EVA</td>
<td>EVA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>EVA</td>
<td>EVA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

A segment of the mission plan can be found below in Figure 5.6, the complete mission plan can be seen in Appendix B.1.

5.4.7 AMSE Cases

In order to evaluate AMSE’s ability to inform mission design and decision making through PHM modelling, several examples of mission crises that may occur were considered and modelled in AMSE. For the purpose of this demonstration of AMSE, it is assumed that these crises were not previously predicted and analyzed. The primary systems of interest for all crises considered are the astronauts and their survival is considered the metric for mission success. Additionally, loss of crew members has the potential to lead to loss of mechanical systems, as it reduces the crew capability to maintain and repair systems, potentially leading to cascading failure. Due to its high
speed AMSE is primarily useful in supporting decision-making in real-time for scenarios that were previously unpredicted or un-modeled.

- **Sol 0**
  - Crew Alpha Arrives on the Surface
  - Perform EVAs and IVAs to verify critical Martian Surface Habitat functionality
  - Unpack transit vehicle
  - Set up habitat module

- **Sol 1-5**
  - Perform EVAs to validate external less critical functions
  - Begin Setup for experimentation and
  - Start farm

- **Sol 6-130**
  - Tend to farm
    - Sol 55: Soybeans mature
    - Sol 67: Wheat mature
    - Sol 75: Potatoes mature
    - Sol 125: Sweet potatoes mature
    - Sol 130: Peanuts mature
    - Sol 130: Self-sufficient food source achieved
  - Perform EVAs on regular schedule
  - Perform IVAs on regular schedule
  - Perform Exercise on regular schedule

*Figure 5.6: Sols 0 through 615 of the General Mission Plan*

**Inaccurate Mission Calculations**

The first crises to be consider in the case study is the response to a faulty assumption or calculation performed in the mission planning stage. Previous robotic missions to Mars have been lost due to incorrect calculations [123]. The example considered is that the estimations for time spent performing tasks is inaccurate and as a result, the expected caloric intake necessary is much lower than the real needs of the astronauts.
In this case, the initial estimate for area to allocate to crops is 35 m², 40 m², 85 m², 65 m², and 4 m² for soybeans, potatoes, peanuts, wheat, and sweet potatoes respectively, in order to serve a caloric demand of 2565 kcal per person per day. However, in reality each astronaut burns 3025 kcal per day in the case study. Crises related to food production and nutrition are of particular interest due to the high impact on mission success and the potentially limited ability to respond due to inability to easily send more food if needed. Additionally, nutrition based crises provided a good test case for AMSE’s ability to model human survival as part of a PHM problem.

Using only the resources available to them on Mars, Crew Alpha must determine a way to compensate for the discrepancy between their available caloric sources and their actual caloric requirements.

*Inability to Farm*

Due to the criticality of food to the mission success [124], a second food inspired case is also considered. In this case, a correct 3025 kcal per day assumption is made during mission planning and enough emergency backup food is planned for triple the time estimated to start the farm and become food self-sufficient (405 Sols). However, due to unknown reasons, none of the crops grow and Crew Alpha must wait for Crew Beta to arrive with more food on Sol 770. With no ability to generate more food, Crew Alpha must explore options to improve their probability of survival using AMSE to inform their decisions.

*Broken Arm*

The mission plan contains many tasks that must be completed and these tasks are initially distributed in order to maximize the probability of mission success. However, there are a wide variety of situations that may necessitate a reallocation of tasks, such as performance of EVAs, to other crew members. This can have potentially dire consequences, because it increases the average caloric load on other astronauts which can lead to nutritional issues as well as increasing the potential exposure to harm, increased wear on assigned EMUs, and increased radiation exposure.

In order to use this class of problems as an example of AMSE’s utility, it is considered that a member of Crew Alpha breaks her arm on Sol 771 when she falls from a ladder in the farm. Analysis using AMSE is performed in order to determine how work should be reassigned in order to give them the necessary time (approximately 70 sols) for their arm to heal with minimal effects on the mission health. Additionally, in order to maintain desired scientific yield and continue to
perform appropriate maintenance actions on mechanical systems work must be reassigned to ensure no EVAs are cancelled.

5.5 Results and Discussion

For each of the cases described above, an initial round of AMSE is performed for the model of the crisis and then options are explored until an acceptable level of mission success is achieved. Acceptable levels of success include situations in which the total probability of mission success over the entire span of the mission does not go below 95% or a case in which no individual’s probability of survival goes below 98% for the mission.

5.5.1 Inaccurate Mission Calculations

For the case of inaccurate mission calculations, a mission plan is created that vastly underestimated the quantity of food that is necessary for survival of the crew. The initial mission plan yields a probability of mission success of 0.5% with the mean probability of survival for each crew member being only 26.6%. The results of the analysis are shown below in Figure 5.7. Over the length of the mission, the average weight of the astronauts’ decreases from 62.50 to 47.99 kg which presents a serious danger from starvation and malnutrition.

Allowing for the possibility that Crew Alpha could use the farm section from Crew Beta’s habitat in order to grow more food, and that Crew Beta can bring along a third farm unit, a solution is found after 1 iteration of AMSE that achieves a probability of mission success of 95.9%. Under this configuration of the mission, 50 m², 60 m², 115 m², 90 m², and 5 m² are allocated for soybeans, potatoes, peanuts, wheat, and sweet potatoes respectively. This plan also allows for all planned work to be continued normally without disruption. The results of the analysis are shown below in Figure 5.8.

5.5.2 Inability to Farm

Similar to the first case, the inability to farm presents a risk from starvation. In this case, only 405-sols worth of rations are brought along to support a 3025 kcal/day diet. Again, Crew Beta is able to adjust what they bring along in order to help solve the problem. However, Crew Beta

2 Neglecting cannibalism.
does not arrive until sol 770, well after the point of starvation if no other mitigating actions are taken. The success and survivability plots for this case are presented in Figure 5.9.

If no action is taken, then the probability of mission success is effectively 0% due to the astronauts starving to death around sol 450\(^3\).

The first option that investigated involves rationing the food to evenly split portions across all 770 sols, which while still insufficient in total calories, at least keeps the food from running out. However, it is found that just rationing the food leads to loss of crew due to starvation sooner due to them being malnourished earlier on by dramatically reducing intake of calories, but not reducing their need caloric usage. The associated plots can be found in Figure 5.10.

AMSE is performed again, and again the reserve of food is rationed to extend available food as long as possible, but all EVAs and exercise are cancelled, and the rest/sleep period is extended from 8.6 hours per day to 16.6 hours per day. While this approach completely halts any planned scientific endeavors, it is enough to keep from dramatic weight loss, and the probability of mission success (defined as keeping the astronauts alive) increases to 92.99% with a mean individual survival probability of 98.2%. This is considered a sufficient solution given the constraints of the problem. The associated plots for this mission plan can be found in Figure 5.11.

One potential consequence of this strategy is that the crew’s ability to respond to additionally crises is severely limited, and taking any actions could potentially lead to starvation. This is compounded by the cancelled EVAs and reduced IVAs, which has numerous effects on the health of physical systems that require scheduled maintenance. For example, when the EVAs are canceled the SEVs are likely to accumulate damage from ordinary Martian weather leading to reduced system health and higher probability of system loss. While the SEVs are not critical to mission survival and their failure does not affect mission success, the potential scientific yield of the mission is limited after rescue by Crew Beta is limited by their loss.

\[ \text{Footnote: This assumes no self-sacrifice such as was depicted in Mission to Mars [125], no cannibalism, and no other extreme solutions.} \]
Figure 5.7: Individual Survival Rate for Inaccurate Caloric Needs
Figure 5.8: Individual Survival Rate for Inaccurate Caloric Needs
Figure 5.9: Individual Survival Probabilities If Unable to Farm
Figure 5.10: Individual Survival Probabilities If Unable to Farm and Food is Rationed
5.5.3 Broken Arm

The broken arm problem investigates what occurs if someone becomes temporarily incapacitated. In this case, astronaut A of Crew Alpha is unable to perform EVAs for 70 sols beginning on sol 771. EVAs are required to be performed by two astronauts at a time in the mission plan in order improve EVA safety. However, if EVAs are cancelled scheduled system maintenance tasks and scientific opportunities are reduced. In order to keep up scientific yield, the EVA schedule is temporarily revised to the one shown in Table 5.4.

This leads to no significant reduction in probability of mission success, with a probability of success of 95.9%. The resulting associated plots can be seen in Figure 5.12.
Figure 5.12: Individual Survival Probabilities If Work Allocation is Change to Allow For Arm to Heal

While this adjustment in task planning doesn’t seem to have a significant influence on the probability of mission success, it does have some effects on the individual astronauts that may result potential consequences. For example, over the course of the mission, astronauts B, C, and D end up being exposed to an additional 0.2 mSV of radiation, which is equivalent to receiving two chest x-rays.
5.5.4 Discussion of Results

In the cases presented above, AMSE is used to make PHM-informed space mission control decisions. In each case, the mission model is reconfigured within several minutes and analysis can be run in under 80 seconds. This allows for rapid response to mission crises. The selected crises for the case study were relatively simple with fairly apparent solutions, but each selected case was representative of a different class of space mission crisis that may be encountered. The selection of simple cases was intentional in order to focus on the demonstration of the AMSE as a method for PHM-informed space mission decision-making.

In initial investigation of the PHM-informed space mission model used for this study, it was found that the probability of mission success was very highly dependent on nutrition of the astronauts, and that maintaining a healthy astronaut and a productive mission would be a difficult balancing act. Additionally, if quantity of work is increased, even temporarily, the caloric load can be thrown greatly out of balance. On Earth this would not be a large problem, because more food can easily be acquired, but on Mars additional food could take several years to arrive as flight times are highly dependent upon launch windows. This observation was part of the inspiration for having multiple cases that focused on food-related crises.

The uniqueness of AMSE in providing a decision support tool that uses real-time system health information to help mission operations managers in rapidly developing unanticipated scenarios positions AMSE to be a useful addition to space missions. The underlying system models that provide risk analysis capability are directly modified by PHM information from the physical systems. In the case of the case study, the systems are simulated; however, we have conducted initial testing on a PHM testbed platform with promising results.

While the case study focused on crises that were relatively easy to avert, the AMSE method is capable of handling much more complicated system failure scenarios. The limiting factor of the AMSE method’s ability to model and analyze a mission is the availability of computational resources and the resolution of the developed mission model.
5.6 Conclusion and Future Work

Active Mission Success Estimation (AMSE), is a method for the modelling and analysis of space missions for the purpose of risk analysis and informed decision making based on real-time PHM information. The bulk of the AMSE method consists of three phases. The first phase of AMSE is modelling. In this phase a functional model containing PHM information is developed of the mission using a system-of-systems approach in order to represent multiple interacting mission components. In addition to the functional model of the system, a mission plan is developed that contains a list of all tasks to be performed over the course of the mission. The tasks are represented by task modules, which contain quantitative information and mathematical models necessary to analyze the effect of the task on the health of systems within the mission framework. The second phase of AMSE, analysis, utilizes the functional model of the system-of-systems and the mission plan in order to perform calculations the determine the probability of mission success over time. This phase is highly dependent on analysis of the system health models developed in Phase 1. The third and final phase of AMSE involves the interpretation of the results of analysis in order to inform mission control decisions.

The AMSE method is shown to be an effective tool for risk-informed PHM-driven space mission control decision making using analysis conducted on functional models representing real systems in a system-of-systems framework. This is demonstrated through the evaluation of three potential crises that could occur during a space mission.

Through the case study, AMSE shows its ability to be rapidly reconfigured in highly detailed ways.

5.6.1 Future Work

AMSE is a promising tool for PHM-informed mission risk analysis and decision making, but is currently limited in its user friendliness and lacks any form of GUI or developed UI and instead relies on the user to make changes to the code performing the analysis. While this is doable it is a non-ideal implementation and it vastly reduces the ability for AMSE to be used by new people. Therefore, development of a GUI for the AMSE code to be run through is given a high priority.

Another area for improvement on AMSE is in the sourcing of functional models which include PHM data and health modelling. Currently models must be developed for each system that
is to be included in the system-of-systems framework. However, a database could be developed of common models for use in AMSE. This would enable for more rapid creation of mission model and improved configurability speeds by allowing for more rapid interchanging of systems or sub-systems.

A final avenue of interest for future investigation is the use of AMSE with an Artificial Intelligence (AI) in order to enable autonomous decision making under risk. Developing better methods for autonomous decision making in hazardous and unknown environments could have applications in a wide variety of fields including, self-driving cars, home robotics, national security, and space exploration.

Acknowledgements

This research was partially supported by United States Nuclear Regulatory Commission Grant Number NRC-HQ-84-14-G-0047. Any opinions or findings of this work are the responsibility of the authors, and do not necessarily reflect the views of the sponsors or collaborators. The authors wish to acknowledge the work of the undergraduate research assistants in the Van Bossuyt lab and specifically wish to thank the following students for their contributions: Alexis Humann, David Hodge, Zachary Mimlitz, and Robin Coleman. The authors wish to thank LeVar Burton, Fred Rogers, Gene Roddenberry, Carl Sagan, and their individual middle school and high school science and technical arts teachers who inspired them to pursue careers in the sciences and engineering, and instilled in them a sense of purpose and compassion.
CHAPTER 6

DISCUSSION AND BROADER IMPACTS

The Conceptual Object-Based Risk Analysis (COBRA) methodology presents a foundation for an improvement in the way that hazard-facing autonomous systems are designed and risk analysis is performed. Through adoption of a Design-Control Integration approach to a system-of-systems problem, improvements can be made to the survivability of systems and more reliable autonomous decisions can be made. Currently, two hardware platforms are under development in order to verify COBRA through real world experimentation including: PHM Enabled Test Rover (PETR), a terrestrial rover, and the Robotic Optimized Bathyscaphe Reliability Testbed (ROBRT), an underwater Remote Operated Vehicle (ROV). PETR and ROBRT were used as exemplar systems for a large quantity of the research performed on COBRA methods.

While the contents of this thesis demonstrates several applications of the COBRA family of methods, more work is to be completed for the integration of the four phases into a single workflow. Chapter 7 goes into further detail on future work and potential improvements to the COBRA methodology. This chapter discusses the insights gained from each the work presented in Chapters 2, 4, and 5 as well as the broader implications of the COBRA method.

6.1 Failure Flow Decision Functions (FFDF)

The work presented in Chapter 4 and the Previous Work section of Chapter 2 on Failure Flow Decision Functions (FFDF) and the application of FFDF for Iterative Design (FFDF-ID) provides a framework for improved autonomous systems to be designed for survivability. This is performed through the analysis of failure propagation through a system, and then the direction of failure away from critical systems and towards less critical systems for improved survivability.

While Chapter 4 focuses on the analysis of a relatively simple functional model of a planetary rover consisting of only 50 functions, the potential exists for application in more complex systems, such as a nuclear power plant or a petroleum refinery, where many redundant flow paths exist for reasons of safety through redundancy or online maintenance. The FFDF methodology allows for better function failure modelling of these and other complex systems while also locating
potential new methods of protecting critical flow exports and functions from failure flows. It is expected that as the FFDF methodology is further refined and fully automated that new and innovative ways of protecting critical flow exports and functions will be discovered that were previously not available to the functional modelling community.

Additionally, it was shown that optimization of design for survivability through FFDF-ID is shown to be an effective method for improving the survivability of a system. The case study and results shows that the FFDF-ID method is able to inform design decisions and lead to a calculated improved survivability of the system.

Several other interesting behaviors are observed in the results. The first notable phenomenon is that in the second phase of the case study, a clear minimum probability of survival emerges in the tradeoff between the store data functions and record visual functions. This phenomenon is observed in all directed and undirected failure cases measured. Upon further investigation, it appears that this configuration has both a very high probability of failing under kNCF conditions and a very high probability of causing failure under ICF conditions. This illustrate the value of the FFDF-ID method by showing that design decisions potential unpredictable results with major consequences on system survivability.

One application of FFDF-ID that may be able to further increase the survivability of a mission is to use FFDF-ID in the design of individuals in a swarm of robots designed to explore an unknown space. In this case, a relatively small increase in individual survivability can lead to a greater survivability of the swarm in its attempt to complete a mission. For example, comparing a swarm 100 robots with the survivability of the initial control design in the case study after one year to a swarm of 100 robots with the final case study design after one year, the probability that at least half of the control swarm will survive is 0.898% and the probability that half the final design swarm will survive is 75.49%. At this scale, FFDF-ID is may be very effective at influencing total mission success.

Another potential application of FFDF-ID is in the analysis of all of the systems involved in a space mission through viewing these systems as a single system-of-systems. Through using FFDF-ID, the survivability of each individual system can be increased, thus leading to a potential large increase in overall mission survivability. As in the case of a swarm of robots, a small increase
to the survivability of each individual system in a system-of-systems may have a large cumulative effect on the total survivability of the mission.

The FFDF-ID method can have positive impacts in many system design efforts where system failure or downtime is highly undesirable including space exploration, transportation, and power generation. For example, a power generation system design approached with FFDF-ID may lead to a design that is more reliable and has higher system up-time which is important for power generation contracts and grid load balancing. FFDF-ID further can also allow for rapid automated design and analysis of an entire power grid including functions for power generation, distribution, and regulation with a focus on continuous and uninterrupted power.

6.2 COBRA-Based Navigational Controls

The work on COBRA-based navigational controls discussed in the Previous Work section of Chapter 2 on the RAIR, PIDAA, and GLPFDIOE methods provides a framework for improved autonomous system controls to be designed for survivability. This is enabled by the performance of COBRA Phase 1 analysis and evaluation of systems was performed in the Simulated Physics and Environment for Autonomous Risk Studies (SPEARs) simulator for autonomous system in a Martian environment.

The RAIR method was shown to have a significant impact on the survivability of a system. One interesting result from this study was that extremely low risk tolerance systems appear to accrue more risk than systems with a moderate risk tolerance, but also managed to reach the target destination more often. This implies that there may be a more complex relationship between the risk tolerance applied to the optimization formula, and the actual risk that accrues. One possible explanation is that there is a region of risk tolerance where a rover is more likely to take a moderate risk, moderate reward action, and that these end up providing less reward over the long run than taking slightly more risk for more reward. Future studies should be performed to better study this relationship.

The PIDAA method was shown to provide an increase in survivability for a simulated rover. Tests from all maps showed a reduction in hazard rate, and as a result the project mission life of the rovers utilizing PIDAA was higher. One problem with the PIDAA method as it was implemented is that the primitive damage mitigation techniques and hazard avoidance which slowed down the rover too much when they encountered a hazard and it was too hesitant to
approach danger. By hybridizing PIDAA with RAIR, a new method to navigate around hazards as they are discovered was developed and became the initial basis for GLPFDOE.

The GLPFDOE method was shown to improve performance of a simulated planetary rover in a Mars-like environment. In addition to utilizing a control scheme based in both PIDAA and RAIR GLPFDOE took a control design integration approach which attempted to merge the development of physical system design with the development of navigational controls. One notable behavior is that the initial design before any iteration was performed scored higher than the first 3 iterations, before improvement in performance was observed. It appears that this may have been a result of taking an evolutionary approach to the design problem and taking an initial step that was non-optimal, and the modification temporality reduced the effectiveness of the system. However, after several iterations the design outperforms the initial design by a factor of approximately 2.5 leading to significantly improved survivability and performance.

6.3 Active Mission Success Estimation (AMSE)

The work presented in Chapter 5 on AMSE provides a framework for performing mission analysis through a functional modelling based approach. This was performed through the development of a functional model of a system-of-systems representing a crewed space mission. While the case study presented is human-centric, the AMSE method works for the analysis of autonomous and semi-autonomous system-centric missions as well, such as a planetary rover.

In initial investigation of the space mission model used for this study, it was found that the probability of mission success was very highly dependent on nutrition of the astronauts, and that maintaining a healthy the model would be a difficult balancing act. Additionally, if quantity of work is increased, even temporarily, the caloric load can be thrown greatly out of balance. On Earth this would not be a large problem, because more food can easily be acquired, but on Mars additional food could take several years to arrive as flight times are highly dependent upon launch windows. The importance of this observation is that it shows the capability of AMSE to direct attention to systems and processes that have a high impact on mission success and system survival. Due to the nature of system-of-systems models the propagation of failure can be highly unpredictable and the ability to determine what is leading to system failure is very valuable for improving design and making informed decisions.
6.4 Broader Impacts

The COBRA family of methods present the potential for significant improvement on the self-sufficiency and survivability of autonomous systems. Through applications of the methods described in Chapters 2, 4, and 5 of this thesis autonomous systems such as Uncrewed Aerial Vehicles (UAV), ROVs, self-driving cars, quadcopters, and assistive home robots for care of the infirm may be improved in their ability to operate in chaotic and uncontrolled environments. Improving autonomous system survivability when facing the unknown is a necessary problem that must be addressed in order for the above listed autonomous systems and other similar systems to operate independently of human intervention.

Focusing specifically on autonomous systems for space exploration, COBRA methods have the potential to vastly improve mission operations. Due to the inherently unknown operating environments of space mission autonomous systems, such as an autonomous or semi-autonomous rover, COBRA methods can have a high impact on improving system survivability. This is enabled through the four phase structure of COBRA. The first stage enables the autonomous system and its designers to better understand the way that failure develops and propagates through the autonomous system. The insight gained in the first phase enables the development of control, navigation, and decision making systems in the second phase. The resulting control-design integration enables systems to react reflexively to potential damage to the system improving survivability. The third phase of COBRA builds upon the first two phases but places the models generated into a model of the operating environment and considers not just the instantaneous system health, but also evaluates the probable effects of decisions made by the autonomous system on mission success. Finally, all of these phases of development can implemented in ways that allow for informed iterative design of the physical hardware systems of the autonomous space exploration system. Through the application of COBRA to autonomous systems for space exploration, the challenges created by communication delay can be overcome and systems that are able to extend human reach and explore the universe without constant human intervention can be created.

While COBRA was designed with the explicit intent of improving autonomous system design for survivability, it has been shown to be adaptable to various other systems as demonstrated in Chapter 5. In addition to the already explored applications for COBRA to crewed space
missions, potential applications exist in the modelling of infrastructure, aerospace vehicles, and terrestrial vehicles. COBRA could improve infrastructure by better modelling sensitivity of power grids or traffic and determining where improvements could be made that would allow for improved reliability and performance of these systems. Additionally, the design and operation of nuclear power facilities could be improved through COBRA and more rapid response the potential crises could be enabled, leading to greater nuclear safety. Both commercial and private aerospace systems could be improved through COBRA by providing improved methods for system designs and informing maintenance schedules that are more efficient without reduction to system safety. For the case of consumer facing vehicles such as cars COBRA enables the design of systems that better monitor their own health and provide expert advice on what maintenance operations are needed and when, leading to greater automotive reliability.
CHAPTER 7
FUTURE WORK

While COBRA has made great strides towards improving autonomous system survivability and providing an improved framework for risk assessment of a variety of systems, continued development on the COBRA methodology should be performed. One of the primary objectives of further developments of COBRA will be the development of a single streamlined methodology that better includes all four phases of COBRA and is capable of performing all of the operations presented in Chapters 2, 4, and 5 of this thesis. A single unified COBRA method would allow for more rapid analysis and more streamlined work by developing a more generalized system model that can then be analyzed in a variety of ways using the same general approach.

In order to enable wider adoption of COBRA and methods in the COBRA family work should be performed on the development of a functional model database that is crowd sourced and contains a variety of individuals functions and groups of functions representative of components, up to models of entire systems or systems of systems. Additionally, parameterized models that are representative of operating environments could be developed, allowing for the analysis of systems under a wider variety of operating conditions, without the need to completely redevelop a system model from scratch. Finally, development of a graphical user interface that would allow for easy assembly of models through the use of the database and a drag and drop interface should be developed. This will enable operations similar to those presented in Chapters 2, 4 and 5 to be performed by individuals with less expertise in the development of system models for risk assessment.

Focusing specifically on the AMSE methodology, further investigation should be performed on the use of an AMSE based methodology for autonomous decision making. The development of Active Mission Success Estimation for Autonomous Decisions (AMSEAD) would enable the analysis of a system-of-systems in an automated manner by automatically generating possible mission decision options and then selecting or recommending options that best improve the probability of mission success. This would enable autonomous systems to better respond to unforeseen circumstances and take failure mitigating actions more quickly, or could be used to
create an advisor that provides insights to a human operator by suggesting courses of action and then effects of total mission success.

One challenge of COBRA is finding effective ways to properly communicate the results of analysis to system operators. Further work should be performed on the exploration of this problem in order to generate reports that take a more human analog approach to communication of data and studies should be performed on what communication methods may be the most effective for encouraging human maintenance of systems and communication of complex technical information. One potential avenue this research could take would be through interpreting the results of analysis of an autonomous system and then using emotive or human analogous health term to draw attentions to systems that may be experiencing issues. For example, instead of a car displaying a message stating “maintenance required” it could instead say something along the lines of “I am not feeling very well today, my fuel consumption has been higher than usual lately and my brake pads hurt”.

Another path that future work could take would be on studying how humans interact with autonomous systems in cooperation. This is important to understand because if humans and autonomous systems such as assistive robots are going to work in the same space, then it is important that the autonomous system be capable of interpreting human actions as part of their decision making model. This line of research could take the form of experiments into machine learning and artificial intelligence studies of human robot interaction through performance of cooperative tasks. By performing a cooperative task such as moving boxes, completion of a puzzle, or playing a simple cooperative game, an understanding of the ways in which humans treat autonomous systems could be developed. The developed understanding could then be used in prediction of how a human cohort would behave and productivity and safety of both the autonomous system and human could be improved.

The proposed future work will be continued by the author of this thesis during the completion of their Doctorate of Philosophy in Mechanical Engineer.
CHAPTER 8
CONCLUSION

Conceptual Object Based Risk Analysis (COBRA) was shown to have a beneficial effect on the design and operation of autonomous systems for survivability. The benefits of COBRA are demonstrated in the research publications presented in Chapters 2, 4, and 5 of this thesis.

Chapter 4: Failure Flow Decision Functions for Iterative Design, explored how COBRA could be used to determine the sensitivity of an autonomous system to failure. This is necessary for improving survivability of autonomous systems, because it provides the basis for informed decision making. This work demonstrates how COBRA can be applied in order to inform early stages of physical design, in addition to informing system control.

The previous work presented in section 2.2.2.1 Risk Attitude Informed Route-Planning (RAIR), explored how COBRA could be applied to make navigational decisions for an autonomous system. Being able to make navigational decisions that improve system survivability is a good demonstration of the ability to make informed decisions in a variety of other contexts, because the navigational problem considers multiple levels of environmental and system information in order to inform decisions. Additionally, the case study performed on RAIR demonstrated several emergent phenomena from the application of COBRA for navigation. The first noticeable emergent behavior of the autonomous system was that systems that were too hesitant to accept dangerous options would converge towards driving away from target destinations and seeking out relatively safe hole to hide in. On the other side of the spectrum, systems that were too accepting of risk had a tendency to get themselves stuck by attempting to take shortcuts that were too hazardous. As a result, an optimal risk attitude emerged for exploration of the simulated environment through acceptance of reasonable risk. This shows that COBRA is capable of shedding light on important design considerations, and that it has the ability to demonstrate behavior unforeseen by the system designers.

The final case study presented in Chapter 5: Active Mission Success Estimation Through PHM-Informed Probabilistic Modelling for a Crewed Martian Mission, demonstrated how COBRA has potential applications outside of autonomous system design and analysis. This is notable for two reasons. The first is that is shows that COBRA has much broader applications than
the systems it was originally designed to analyze. The second is that COBRA being able to model and understand non-mechatronic systems, such as humans, is critical because the autonomous systems being designed will not always be operating in a vacuum, and the ability to understand how the autonomous systems actions may impact the health of a human is critical for making informed decisions.

In conclusion Conceptual Object Based Risk Analysis (COBRA) provides not only the potential to improve the design of autonomous systems for survivability, but also the potential to improve the reliability and safety of a wide variety of systems.
REFERENCES


[38] Squyres, S. W., Knoll, A. H., Arvidson, R. E., Clark, B. C., Grotzinger, J. P., Jolliff, B. L., McLennan, S. M., Tosca, N., Bell, J. F., Calvin, W. M., and others, 2006, “Two years at Meridiani Planum: results from the Opportunity Rover,” Science, 313(5792), pp. 1403–1407.


110


Pecht, M., 2008, Prognostics and health management of electronics, Wiley Online Library.


Jensen, D., Tumer, I. Y., and Kurtoglu, T., 2015, “FLOW STATE LOGIC (FSL) FOR ANALYSIS OF FAILURE PROPAGATION IN EARLY DESIGN.”


APPENDIX A.1

Collectable Energy
Electrical Energy
Digital Signal
Control Signal
Position Information
Visual Information
Rotational Information
Translation Work
Steering Work
Warm Electronics
Box Space
## APPENDIX A.2

<table>
<thead>
<tr>
<th>Flow Type</th>
<th>Probability of Passing Failure Downstream</th>
<th>Probability of Passing Failure Upstream</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collectable Energy</td>
<td>0.20</td>
<td>0.00</td>
</tr>
<tr>
<td>Electrical Energy</td>
<td>0.80</td>
<td>0.02</td>
</tr>
<tr>
<td>Digital Signal</td>
<td>0.99</td>
<td>0.02</td>
</tr>
<tr>
<td>Control Signal</td>
<td>0.99</td>
<td>0.02</td>
</tr>
<tr>
<td>Positional Information</td>
<td>0.95</td>
<td>0.00</td>
</tr>
<tr>
<td>Visual Information</td>
<td>0.95</td>
<td>0.00</td>
</tr>
<tr>
<td>Rotational Work</td>
<td>0.99</td>
<td>0.15</td>
</tr>
<tr>
<td>Translational Work</td>
<td>0.99</td>
<td>0.15</td>
</tr>
<tr>
<td>Alignment Work</td>
<td>0.50</td>
<td>0.15</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Function</th>
<th>Probability of Accepting Failure Flow</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accumulate Energy</td>
<td>0.50</td>
</tr>
<tr>
<td>Store Energy</td>
<td>0.12</td>
</tr>
<tr>
<td>Distribute Electrical</td>
<td>0.24</td>
</tr>
<tr>
<td>Control Magnitude Electrical</td>
<td>0.20</td>
</tr>
<tr>
<td>Convert Electrical to Rotation</td>
<td>0.16</td>
</tr>
<tr>
<td>Transmit Rotation</td>
<td>0.16</td>
</tr>
<tr>
<td>Convert Rotation to Translation</td>
<td>0.16</td>
</tr>
<tr>
<td>Direct Command</td>
<td>0.44</td>
</tr>
<tr>
<td>Process Signal</td>
<td>0.01</td>
</tr>
<tr>
<td>Store Data</td>
<td>0.01</td>
</tr>
<tr>
<td>Record Position</td>
<td>0.22</td>
</tr>
<tr>
<td>Record Visual</td>
<td>0.22</td>
</tr>
<tr>
<td>Transmit Data</td>
<td>0.44</td>
</tr>
</tbody>
</table>
APPENDIX A.3

Collectable Energy
Electrical Energy
Digital Signal
Control Signal
Position Information
Visual Information
Rotational Work
Translation Work
Steering Work
Direct Failure

Warm Electronics
Box Space

Direct Failure
APPENDIX B.1

Mission Plan

• Sol -770
  o Crew Alpha Equipment Arrives on Planet
• Sol 0
  o Crew Alpha Arrives on the Surface
  o Perform EVAs and IVAs to verify critical Martian Surface Habitat functionality
  o Unpack transit vehicle
  o Set up habitat module
• Sol 1-5
  o Perform EVAs to validate external less critical functions
  o Begin Setup for experimentation and
  o Start farm
• Sol 6-130
  o Tend to farm
    ▪ Sol 55: Soybeans mature
    ▪ Sol 67: Wheat mature
    ▪ Sol 75: Potatoes mature
    ▪ Sol 125: Sweet potatoes mature
    ▪ Sol 130: Peanuts mature
    ▪ Sol 130: Self-sufficient food source achieved
  o Perform EVAs on regular schedule
  o Perform IVAs on regular schedule
  o Perform Exercise on regular schedule
• Sol 131-615
  o Perform EVAs on regular schedule
  o Perform IVAs on regular schedule
  o Perform Exercise on regular schedule
• Sol 616-620
  o Begin verification of Beta Martian Surface Habitat during EVAs
  o Perform IVAs on regular schedule
  o Perform Exercise on regular schedule
• Sol 621-769
  o Sol 621
    ▪ Begin Farm Beta
  o Tend to Farm Beta
    ▪ Sol 671: Soybeans mature
    ▪ Sol 688: Wheat mature
    ▪ Sol 696: Potatoes mature
    ▪ Sol 746: Sweet potatoes mature
    ▪ Sol 751: Peanuts mature
    ▪ Sol 751: Self-sufficient food source achieved
  o Perform EVAs on regular schedule
  o Perform IVAs on regular schedule
• Sol 770
  o Crew Beta Arrives on surface
    ▪ Perform EVAs and IVAs to verify critical habitat functionality
    ▪ Unpack transit vehicle
    ▪ Set up habitat module
• Sol 771-775
  o Crew Alpha
    ▪ Perform EVAs on regular schedule
    ▪ Perform IVAs on regular schedule
    ▪ Perform Exercise on regular schedule
  o Crew Beta
    ▪ Perform EVAs to validate external less critical functions
    ▪ Begin Setup for experimentation and
• Sols 776-1050
  o Crew Alpha
    ▪ Perform EVAs on regular schedule
    ▪ Perform IVAs on regular schedule
    ▪ Perform Exercise on regular schedule
  o Crew Beta
    ▪ Perform EVAs on regular schedule
    ▪ Perform IVAs on regular schedule
    ▪ Perform Exercise on regular schedule
• Sols 1051-1069
  o Crew Alpha
    ▪ Begin Prep for departure
    ▪ Wrap up experiments
    ▪ Perform EVAs to hand off tasks to Beta
    ▪ Prepare habitat Alpha for vacancy
      • Will be used by Crew Gamma
  o Crew Beta
    ▪ Perform EVAs on regular schedule
    ▪ Perform IVAs on regular schedule
    ▪ Perform Exercise on regular schedule
• Sol 1070
  o Crew Alpha Departs from Martian Surface
  o End AMSE analysis
• Sol 1540
  o Crew Gamma arrives and moves into habitat Alpha