Development of BDD Models for Decision Support in Phased Mission Systems

Yang Zhang

Thesis submitted to The University of Nottingham

for the degree of Doctor of Philosophy

March 2016

The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

Abstract

Autonomous systems are becoming increasingly commonplace, with applications either existing or suggested in many different industries. As levels of autonomy increase, the need for these systems to interpret with environments in which they operating and make decisions about their own future actions following internal failures or external threats. In the past, reliability analysis methods have been suggested as having the potential to provide information that could be used in a real-time decision support tool for autonomous systems in changing environments. Real-time support is particularly important in systems such as unmanned aerial vehicles (UAV), where any delay in making a decision following a failure occurrence or the emergence of a threat could be catastrophic.

Reliability Analysis can be used to calculate the failure probability of a mission such as that performed by a UAV by modelling the mission as a sequence of tasks known as a phased mission.

Binary Decision Diagram models have shown great potential for analysing phased mission systems since they can produce accurate mission and phase failure probabilities in reasonably short time frames. Although research to date has shown that Binary Decision Diagrams appear to have the most promise for performing the real-time analysis that would be required as an input to a decision making tool for phased mission systems, the analysis as it stands still falls some way short of being near-instant, as it must be for decisions to be made quickly when required. In common with many systems, phased mission systems can contain components that fail in multiple failure modes. It is therefore important that multiple failure modes are modelled while developing the Binary Decision Diagram tools and techniques considered in this research.

The research presented in this thesis aims to address the deficiencies seen in previous methods by investigating the Binary Decision Diagram techniques and suggesting how the techniques can be developed for use within a decision support tool where fast, accurate decision making is required. The novelty of the research is as follows:

1. Different Binary Decision Diagram models for phased mission systems are reviewed and three new Binary Decision Diagram models are proposed to improve the efficiency and accuracy of analysis for phased mission systems containing multiple failure mode components.

2. Since the size of a Binary Decision Diagram has a significant effect on the time required to quantify it and the Binary Decision Diagram size is influenced by variable ordering, nine different variable ordering schemes are investigated for phased mission systems. Eight of them are extended from fault tree analysis of single phase systems containing single failure mode components and one is newly-developed specially for use within a decision support tool.

3. Due to the potential time limitation for decision making, approximation methods are investigated to evaluate the failure probabilities in phased mission systems while trading off between accuracy and analysis efficiency. Three new approximation models are developed and their analysis efficiency advantage over the exact analysis is demonstrated testing on a large number of sample phased mission systems. A performance indicator is developed in order to facilitate the choice of approximation method taking into account accuracy and efficiency requirements.

The benefits of the developed methods are demonstrated through the consideration of a case study.

Acknowledgements

First of all, I would like to express my sincere thanks and appreciation to my supervisors Dr Darren Prescott and Prof John Andrews, for their continuous support, motivation and guidance, for their inspiration, immense knowledge and advice, for their patience and tolerance throughout my PhD research. I am very grateful to them for providing me the chance to learn and work as a PhD student in the Risk and Reliability Group.

I would also like to express my thanks to Dr Rasa Remenyte-Prescott, for her guidance and the discussions we had to address my confusions during the research. I would also like to thank Dr Sean Reed, for his expertise and help in the coding area. I would like to give thanks to my officemates Claudia Fecarotti, Dr Marius Vileiniskis, Hector Reyes, Jack Reeves, and Dr Jingyu Sheng, Serena D'souza, for their companionship in my tough thesis-writing period. I would like to thank all the current and previous members I have met in the research group for their continuous support and friendship. Thanks to all the friendly people in the Nottingham Transportation Engineering Centre who have made my PhD life more enjoyable away from my studies.

Finally, I would like to thank my family and friends for their constant support and encouragement throughout the research. Special thanks to the one, Dr Jiarui Cao, for his consistent tolerance, patience, concern and counselling, without whose support, I would not have gone through this event.

Contents

Notations

$+$ Boolean operation $Union$, equivalent to OR gate in a fault tree, page 23

$\bar{x}$ Boolean representation *complementation* of variable x, equivalent to NOT gate in a fault tree, page 23

$\cdot$ Boolean operation *intersection*, equivalent to AND gate in a fault tree, page 23

$\Diamond$ Boolean operator, $AND(+)$ or $OR(\cdot)$, of the logic gate in the fault tree., page 30

$\overline{A_{0i}^{p}}$ Component A does not fail in failure mode p between the start of the mission and the end of phase i, page 77

$\overline{F_i}$ The success logic of phase i, page 45

A_{0i}^{p} Components A fails in failure mode p between the start of the mission and the end of phase i, page 77

A_{mn} The failure of component A between the end of phase m and the end of phase n, page 40

$cp(x)$, $fm(x)$ and $pn(x)$ The component to which x relates, its failure mode and the indices of the phases within which x occurs, respectively., page 79

$F_1 + F_2 + \cdots F_i$ A system failure having occurred at some point between the start of the mission and the end of phase i, page 40

F_i Failure logic of phase i, page 39

F_{miss} Mission failure logic, page 39

$P(ph_i)$ Conditional unreliability of phase i, page 40

ph_i logic expression for the conditional failure of phase i, i.e, a system failure has occurred on phase i conditional on the success of the previous $i-1$ phases, page 40

Q_{miss} The mission unreliability, page 40

$*$ Multiplication of probabilities, page 145

BDD Binary Decision Diagrams, page 13

Component of γ The component to which the variables in γ relate, page 151

Contradiction task The task which once selected, the considered task fault tree module no longer being a module for the phased mission fault tree, page 59

PMS Phased Mission Systems , page 37

UAV Unmanned aerial vehicles, page 1

Chapter 1

Introduction

1.1 Background and Research Motivation

Autonomous systems are increasingly used in many different industries, with Unmanned Aerial Vehicles (UAVs), Mars rovers, and driverless cars being just three examples that have been widely reported in recent times. In order to be totally autonomous, these systems must be capable of interpreting their environment and making decisions on their future behaviours without the need for human intervention. The reaction of many systems, autonomous or otherwise, can require them to perform phased missions. Phased missions are those in which systems are required to perform a number of different tasks in sequence. An example of such a mission is an aircraft flight, which includes phases of taxiing to the runway, taking off, climbing to a cruising altitude, cruising, descending, landing and taxiing back to the terminal.

In order to successfully complete a phased mission, a system must manage to complete each of the individual phases of that mission without failure. If internal failures or external threats occur, the system may need to be reconfigured in order to ensure the mission can still be completed or there may be a need for alternative missions to be considered to guarantee the system does not experience a failure leading to severe, catastrophic consequences. A key factor in making decisions in a changing mission environment is the probability of successful operation through future mission phases and the mission as a whole. Fast, accurate calculations of the probability of failure to complete future phases and the entire mission could therefore form a crucial element of a decision making process, within autonomous systems and systems that are controlled by human operators. If the probability of failure leads to the risk associated with any proposed mission reaching an

unacceptable level, then the risk associated with other mission configurations must be considered in order to identify an acceptable alternative. In order to be able to evaluate the options available and select the best possible course of action in a timely manner in a dynamic, rapidly-changing environment where sound decisions must be made quickly, the fast, accurate calculation of phase and mission failure probabilities is critical.

In a phased mission, the mission configurations, failure criteria and states of system components can vary between different phases, since the objectives of each of the individual phases are different. Each phase is identified by a phase index, phase length, and failure criteria. A component may fail in any phase in the mission and may or may not contribute to the failure of the system to complete that phase. However, while the failure of a component may not immediately lead to system failure, it may still contribute to the failure of the system in a later phase; for example, consider a failure occurring to an aircraft landing gear component during the cruise phase that does not lead to system failure until the aircraft comes to land. The phased mission system reliability is defined as the probability that the system operates successfully in all of its phases. The phased mission system unreliability is defined as the probability that it fails to complete at least one phase.

The phased mission systems considered in this research are non-repairable, that is, if one component fails, then the systems will remain failed for the rest of the mission. One important feature of such phased mission systems is the dependency that exists between phases and within components. A component or subsystem could have more than one failure mode. Once the component has failed in a certain failure mode, it will remain in the failed state for the rest of the mission. For example, a switch in an engine system could fail to open or fail to close. If the switch fails to open in one phase, then it will fail to open in all following phases and it will not be able to fail to close either. This feature of phased mission systems introduces dependencies between components and phases, which make the analysis significantly more complex than that of single phase systems.

Previous research has attempted to analyse non-repairable phased missions using techniques such as fault tree analysis. Fault tree analysis is commonly used in industry for calculating the probability of system failure. It assumes that all components in the system are independent, which simplifies the analysis but leads to inaccuracies, since there often exist dependencies between different phases and often within the same phase, e.g. in the case of components with multiple failure modes. The traditional quantification method

requires the identification of minimal cut sets before quantitative analysis is performed and can be extremely computationally intensive, with full analysis often being impossible to implement [28]. This means that the use of traditional reliability quantification techniques such as fault tree analysis would not be appropriate for use in a real-time decision support tool for systems performing phased missions.

1.2 Research Aim and Objectives

The aim of this research is to develop appropriate models to analyse the reliability of phased mission systems quickly and accurately, with a view to providing a reliability analysis methodology that could be used in a real-time decision support tool for systems operating phased missions in changing mission environments. Such a tool could be used in autonomous systems to make timely decisions on the best next course of action when they experience an internal failure or external threat that affects their chance of completing their planned mission. Such tools can also be used as aids to human operators of a system that experiences similar changing circumstances. If a reliability analysis methodology can be developed to allow fast, accurate phased mission quantification, then the reliability of a number of alternative mission scenarios can be calculated, in addition to the reliability of the originally planned mission, allowing a comparison of the relative chances of success and an informed decision to be made as to the best next course of action. It is critical that the models developed allow fast, accurate phased mission analysis, but when the time available to make a decision is limited, it may also be necessary to be able to make a trade off between the accuracy and speed of the calculation performed.

To fulfil the research goal, several aspects must be considered.

- For non-repairable phased mission systems, fault tree analysis is commonly used for the reliability analysis of practical systems. Binary Decision Diagrams (BDDs) [33] can greatly improve the efficiency and accuracy of fault tree analysis compared with the traditionally used kinetic tree theory. Therefore, the first objective of the research is to review the fault tree analysis and BDD approaches.

- Phased mission analysis is far more complex than the analysis of missions with single phases because of the involvement of the dependency between phases and within components. The second objective is thus to review existing BDD models for phased mission analysis.

- The next objective is to improve the efficiency and accuracy of existing BDD models for phased mission systems, to compare their performance by testing on a large number of systems and then to provide recommendations on the selection of BDD models based on different scenarios.

- Since the size of a BDD has a major impact on the time taken to quantify it, and that the order of variables when constructing the BDD influences the size of a BDD, variable ordering schemes for phased mission systems will be investigated in order to allow minimisation of the BDD size and therefore speed up the phased mission analysis.

- Due to the requirement for real-time analysis and the potentially limited amount of time available for decision making, approximations might be needed in order to carry out reliability analysis of mission alternatives in a reasonable time frame. Different approximation methods will be researched and compared in order to choose an optimal model under the required circumstances.

- The developed reliability analysis techniques must be shown to be fit for use in a decision support tool. Therefore, the application of the developed phased mission analysis methods will be demonstrated by the consideration of an example search and rescue mission involving a UAV platfrom, where circumstances require mission reconfiguration to take place.

1.3 Thesis Structure

Chapter 2 reviews the reliability tools that will be needed in order to achieve the research objectives. It begins with a brief overview of risk and reliability theories, and followed by an introduction to qualitative and quantitative fault trees analysis, an introduction to fault tree analysis using BDDs, a review of phased mission analysis using fault trees and a discussion of how phased mission analysis can be used as a part of a decision making tool for autonomous systems.

Since the whole research focuses on phased mission analysis using BDDs, chapter 3 provides a detailed literature review of BDD models for phased mission analysis. Two different modelling approaches, one considering phase dependency during BDD construction and the other considering phase dependency during quantification, are reviewed and their advantages and disadvantages critically appraised.

In chapter 4, two BDD models for phased missions analysis with multiple failure modes are reviewed. Two BDD construction modifications are proposed to make the BDD model developed in [43] more accurate. A novel quantification method is proposed to expedite the quantitative analysis of the BDD constructed in [36]. The analysis methods are followed by comparisons and discussion of tests carried out on a large number of sample systems.

Since variable ordering affects the size of BDDs and therefore their analysis time, variable ordering schemes for phased mission systems are investigated in chapter 5. First, eight variable ordering schemes applied during standard fault trees analysis are reviewed. These are then extended to allow variable ordering in phased mission fault trees. A novel variable ordering scheme using an interleaving technique is then proposed. Finally, the nine ordering schemes are tested on a number of systems and their performances compared.

Chapter 6 details research into the approximation models for phased mission systems. Three BDD-based approximation methods are developed. One is based on the early stage approximation method where phase dependency is not considered, another obtains a conservative evaluation of mission unreliability by constructing BDDs whose root nodes give the rare event approximation of top event probability; the final method provides mission unreliability upper and lower bounds of the mission unreliability by constructing truncated BDD according to specified limits. The methods are tested and discussed.

Chapter 7 describes a scenario, which is then used to demonstrate the capability of the developed BDD models for phased mission systems with multiple failure modes, the impact of the variable ordering schemes and the developed approximation methods.

Chapter 8 summarises the whole research project, highlights the research contribution, and gives recommendations for future research.

Chapter 2

Reliability Background

2.1 Risk and Reliability Definitions

Reliability and risk assessment methods have rapidly developed since World War I and are now widely used in many industries, such as the aircraft, nuclear power and space industries. Industrial engineering systems are not perfect, which means failures cannot be avoided. A failure occurs when the system can no longer satisfy its functional requirements due to internal or external conditions exceeding the system's inherent capacity.

The measure of system performance can be used to help to decide whether or not the risks associated with a system are acceptable. For risk assessment, $Risk$, is generally defined as the product of the consequences of a particular incident, C, and its probability over a time period or the frequency of its occurrence, P:

$$Risk = C \cdot P\,. \tag{2.1.1}$$

Therefore, risk can be reduced by minimising the consequences of the incident C, or by reducing the probability or the frequency of its occurrence, P. Since no system can be risk-free or function eternally, engineers try to minimise risk and ensure the risk associated with a system is acceptable. The acceptability will be judged against criteria related to the safety and economic performance of the system.

It is often impossible to determine when a component or a system will fail. Therefore, stochastic models, which make use of probability theories to quantify the probability of event occurrence, are used. One of the most commonly-used probabilistic measures is system reliability.

Reliability, $R(t)$, is a measure of the probability of successful performance of a system

over a period of time. It can be defined as the probability that an item, such as a component, a piece of equipment or a system, will operate without failure for a stated period of time under specified conditions. The *unreliability*, $F(t)$, is defined as the probability that a system failure has occurred during the period $[0, t)$ given that the system was working at $t = 0$. Since reliability and unreliability are complementary,

$$F(t) + R(t) = 1. \tag{2.1.2}$$

Hazard rate, $h(t)$, is defined for non-repairable system as the rate of failure during the next instant of time for the system has functioned from time 0 to t. If $f(t)$ is the probability density function for $F(t)$, i.e, $\int_0^t f(\tau)d\tau = F(t)$, $h(t)$ is defined as:

$$h(t) = \frac{\text{P[system fails in } [t, t + dt]]}{\text{P[system is functioning on } t]} = \frac{f(t)dt}{R(t)}. \tag{2.1.3}$$

When the hazard rate is a constant, it is frequently called the failure rate and denoted as λ [2].

2.2 Fault Tree Analysis

Fault tree analysis is a commonly-used reliability analysis method [2]. Since it was originally developed in 1962 at Bell Laboratories by H.A. Watson [34], fault tree analysis has been widely used in failure analysis and reliability assessment. The fault tree is a deductive failure model, where a specific system failure mode is expressed as a combination of lower-level modes using Boolean logic.

2.2.1 Fault Tree Construction

Fault trees contains two basic elements: gates and events. The specific system failure state to be analysed is called the *top event*, and the fault tree is developed by determining the immediate, necessary, and sufficient causes, which are expressed by lower-level events. These lower-level events are then further decomposed into lower resolutions until terminating events are encountered. The terminating events, where no further decomposition can be made, at the bottom of a fault tree branch, are known as *basic events* and are usually related to component failure states. Basic events relating to different components are usually assumed to be independent in order to allow quantitative analysis to be performed. The lower-level events are called intermediate events. Gates show the logic that

determines how the input events combine to cause the occurrence of their output, higher level events. The event and gate symbols used in this research are shown in Table 2.2.1.

	Event symbol	Meaning of the symbol
1		Top event or intermediate event
2		Basic event

	Gate symbol	Gate name	Meaning of the symbol
1		AND gate	Output event occurs if all input events exist simultaneously
2		OR gate	Output event occurs if at least one of the input events exists
3		NOT gate	Output event occurs if the input events do not exist

Table 2.2.1: Fault trees symbols

2.2.1.1 Example

For the fault tree shown in Figure 2.2.1, basic events $X1$, $X2$ and $X3$ are independent to each other, the top event is an AND combination of two gate inputs, $G1$ and $G2$, which are OR combinations of events $X1$, $X2$ and events $X2$, $X3$, respectively. After the

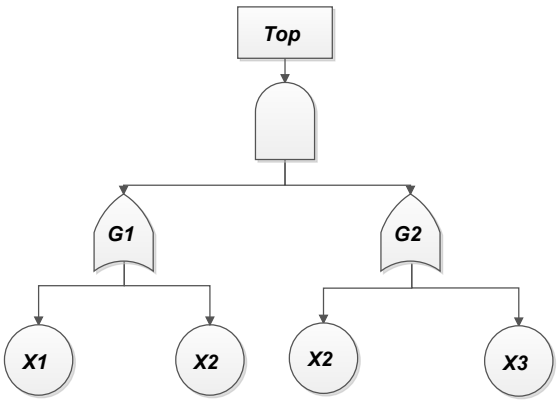

Figure 2.2.1: A fault tree example

fault tree for a system failure mode is obtained, two types of analysis can be performed: qualitative analysis and quantitative analysis. Qualitative analysis involves the identifying the minimal combination of basic events that lead to system failure whereas quantitative analysis involves calculation of system failure parameters including top event probability, top event frequency and event importance measures.

2.2.2 Fault Tree Modularisation

Analysis of large fault trees can be computationally expensive. One way to tackle this problem is to use the modularisation technique. The modularisation technique detects modules, or subtrees, which are completely independent from the rest of the events in the fault tree. That is, a module does not contain any basic event that appears elsewhere in the fault tree. Identified modules can be replaced by module events. The advantage of identifying these modules is that each of them can be analysed separately from the rest of the fault tree, which is relative easy, and then these analysis results can be substituted into the higher-level fault tree where the modules occur to analyse the entire fault tree.

2.2.2.1 Linear-time Algorithm

Fault tree modules can be detected by a linear-time algorithm [13], which detects modules efficiently using only two depth-first traversal of the fault tree. 'Depth-first' means events and gates under the output gate are visited first rather than other events or gates that are parallel with the output gate. The first traversal goes step-by-step through each gate and event and records the steps and the step numbers at the first, second and final visits to the node. The second traversal passes through the tree to find the maximum of the last visits and the minimum of the first visits to the descendants of each gates.

The principle of the algorithm is that if any descendant of a gate has a first visit step number smaller than the first visit step number of the gate, then it must also occur beneath some other gate. Conversely, if any descendant has a last visit step number greater than the secondary visit step number of the gate, then again it must occur elsewhere in the tree. Therefore, a gate can be identified as a module only if the following conditions are satisfied:

- The first visit to each descendant is after the first visit to the gate.
- The last visit to each descendant is before the secondary visit to the gate.

2.2.2.2 Modularisation of an Example Fault Tree

To demonstrate the modularisation process, consider the fault tree in Figure 2.2.2. Beginning with the top event and passing through the tree in a depth-first manner, the visiting number of each gate and event are shown in Table 2.2.2.

Event inputs to any gates are considered before the gate inputs as demonstrated in

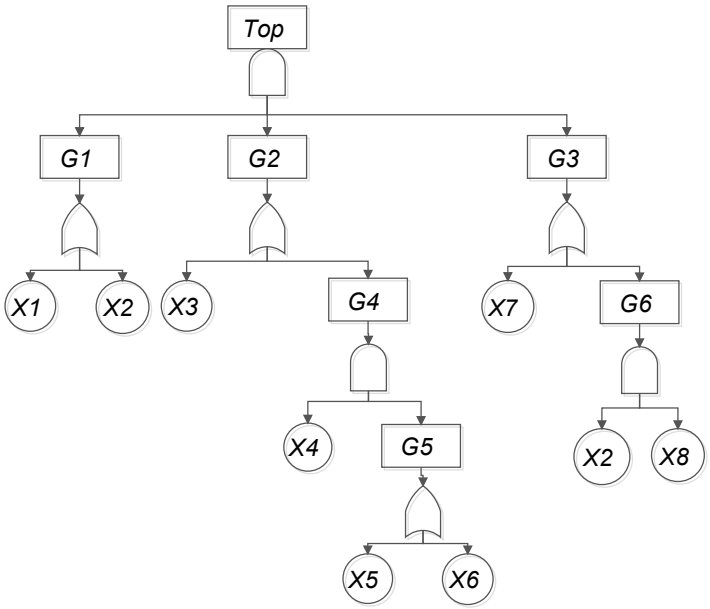

Figure 2.2.2: An example fault tree to illustrate the modularisation technique

Step Number	1	2	3	4	5	6	7	8
Node	Top	G1	X1	X2	G1	G2	X3	G4
Step Number	9	10	11	12	13	14	15	16
Node	X4	G5	X5	X6	G5	G4	G2	G3
Step Number	17	18	19	20	21	22	23	
Node	X7	G6	X2	X8	G6	G3	Top	

Table 2.2.2: Order in which the gates and events are visited in the depth-first traversal of the fault tree in Figure 2.2.2

Gates	Top	G1	G2	G3	G4	G5	G6	Events	X1	X2	X3	X4	X5	X6	X7	X8
1st visit	1	2	6	16	8	10	18	Visit 1	3	4	7	9	11	12	17	20
2nd visit	23	5	15	22	14	13	21	Visit 2	3	19	7	9	11	12	17	20
Last Visit	23	5	15	22	14	13	21	Last Visit	3	19	7	9	11	12	17	20
Min	2	3	7	4	9	11	4									
Max	22	19	14	21	13	12	20									

Table 2.2.3: Critical step numbers for gates and basic events appear in Figure 2.2.2

the table. Each gate is visited at least twice: once on the way down the tree and the other on the way back up the tree. A gate that has been visited can be visited again, but the depth-first traversal will not repeated.

The second pass through the tree will find maximum of the last visit and the minimum of the first visit step numbers of the descendants of each gate and the first visit, second visit and last visit step number for each event, these step numbers are shown in Table 2.2.3.

For this fault tree, gates $G2$, $G4$, and $G5$ are all modules since each descendant under $G2/G4/G5$ has first visit step number bigger than the first visit step number of those gates and has last visit step number smaller than the secondary visit step number of the gates. The top event of a fault tree is always a module (thus there is no need to consider it as an extra module event), since it is always the first event to be visited first and the last to be visited a second time so the two conditions are always applies. Gate $G1$, $G3$ and $G6$ are not modules as they do not satisfy the conditions of being modules. Therefore, gates $G2, G4$, and $G5$ are identified as modules and are replaced by single module events, denoted as $G5 - M1, G4 - M2, G2 - M3$. The modules and the fault tree after modularisation are shown in Figure 2.2.3. Once modules have been found, each of them can be analysed separately and substituted into the original fault tree. This technique significantly reduces the amount of calculation required in the subsequent analysis, as will be demonstrated in the analysis of an example system presented in Section 2.5.3.2.

2.2.3 Qualitative Analysis

Every system failure is caused by some combination of component failure modes. Qualitative analysis is carried out in order to find out which combinations of component failures can cause the system to fail.

A *cut set* is a collection of basic events such that if they all occur then the top event also occurs. However, fault trees for industrial systems can contain large numbers of cut sets which themselves can contain large numbers of basic events. It is more efficient to

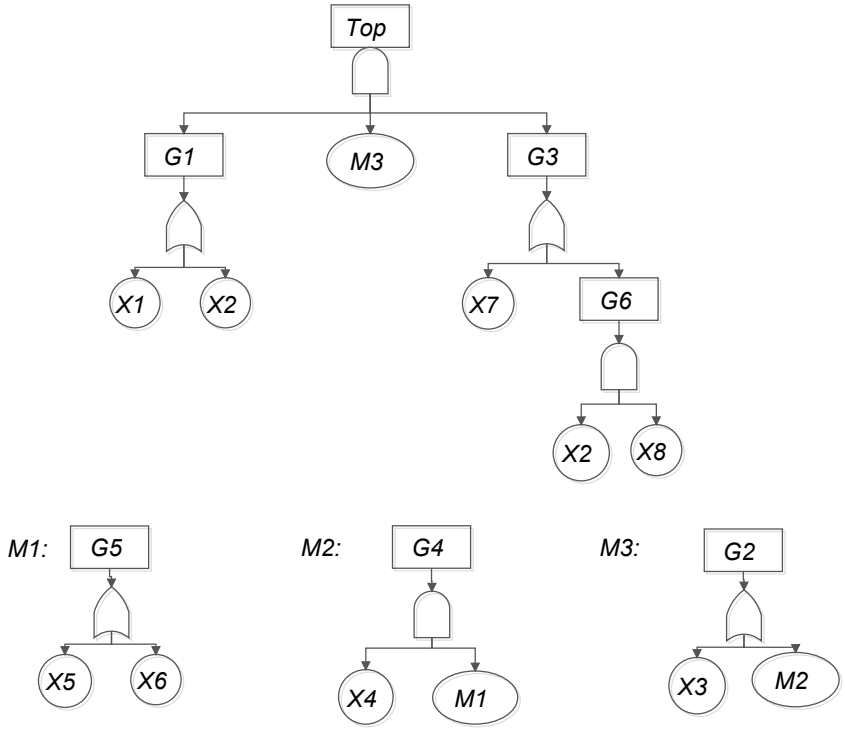

Figure 2.2.3: The four modules obtained from the fault tree in Figure 2.2.2

focus on the *minimal cut set*, defined as the smallest combination of basic events whose failure can lead to system failure, i.e, if any basic event is removed from the set, the top event will no longer occur [26]. Once the minimal cut sets are identified, quantitative analysis can be carried out.

One fault tree can have several minimal cut sets, and if two fault trees have identical minimal cut sets, then they are logically equivalent.

The minimal cut set expression for the top event is:

$$Top = C_1 + C_2 + \cdots + C_n \tag{2.2.1}$$

where C_i, $i = 1, 2, \cdots, n$ is the minimal cut set and each minimal cut set is an AND combination of basic events, which can be expressed by:

$$C_i = X_1 \cdot X_2 \cdot \cdots \cdot X_k \tag{2.2.2}$$

where X_i, $i = 1, 2, \cdots, k$ are independent basic events.

A fault tree whose minimal cut sets are expressed only in terms of component failure modes is known as a *coherent fault tree*, while a fault tree whose minimal cut sets are

expressed in terms of both component failure modes and success modes is a *non-coherent fault tree*. In non-coherent systems, the combinations of basic events which lead to the occurrence of top events are known as *implicants*; the minimal combinations of component failure and success states that lead to the occurrence of the top events are called *prime implicants*. The implicants are equivalent to cut sets in coherent systems and the prime implicants are equivalent to minimal cut sets [2].

2.2.3.1 Boolean Laws

The determination of a fault tree's cut sets or minimal cut sets involves the Boolean manipulation of the top event expression. Basic events are treated as Boolean variables. The three Boolean operations *union* $(+)$, *intersection* $(\cdot)$, and *complementation* $(\bar{x})$ combine Boolean variables in exactly the same way as the three fundamental gates OR, AND, NOT combining basic events. Suppose x is a Boolean variable, and the values 1 and 0 are used to represent its two states:

$$x = \begin{cases} 1 & \text{for event occurrence,} \\ 0 & \text{for event non-occurrence.} \end{cases} \tag{2.2.3}$$

$\bar{x}$ is $NOT\ x$, which is the same as the complement. The rules of Boolean algebra for two Boolean variables x and y are shown below:

1. Commutative laws:

$$x + y = x + y\,, \tag{2.2.4}$$

$$x \cdot y = y \cdot x\,. \tag{2.2.5}$$

2. Associative laws:

$$(x + y) + z = x + (y + z)\,, \tag{2.2.6}$$

$$(x \cdot y) \cdot z = x \cdot (y \cdot z)\,. \tag{2.2.7}$$

3. Distributive laws:

$$x + (y \cdot z) = (x + y) \cdot (x + z)\,, \tag{2.2.8}$$

$$x \cdot (y + z) = x \cdot y + x \cdot z\,. \tag{2.2.9}$$

4. Identities:

$$\begin{aligned} x + 0 &= x\,, \qquad x + 1 = 1\,, \\ x \cdot 0 &= 0\,, \qquad x \cdot 1 = x\,. \end{aligned} \tag{2.2.10}$$

5. Idempotent law:

$$x + x = x\,, \tag{2.2.11}$$

$$x \cdot x = x\,. \tag{2.2.12}$$

6. Absorption law:

$$x + x \cdot y = x\,, \tag{2.2.13}$$

$$x \cdot (x + y) = x\,. \tag{2.2.14}$$

7. Complement:

$$\begin{aligned} x + \bar{x} &= 1\,, \\ x \cdot \bar{x} &= 0\,, \\ \overline{(\bar{x})} &= x\,. \end{aligned} \tag{2.2.15}$$

8. De Morgan's laws:

$$\begin{aligned} \overline{(x + y)} &= \bar{x} \cdot \bar{y}\,, \\ \overline{x \cdot y} &= \bar{x} + \bar{y}\,. \end{aligned} \tag{2.2.16}$$

Minimal cut sets are obtained by applying the Boolean laws to the Boolean expression for the fault tree top event. The laws are applied until the expression is in a sum of products form, where each product contains variables independent of each other.

2.2.3.2 Example

Consider for example the fault tree shown in Figure 2.2.1. The Boolean expression for the two intermediate gates, $G1$ and $G2$, can be formulated as:

$$\begin{aligned} G1 &= X1 + X2\,, \\ G2 &= X2 + X3. \end{aligned} \tag{2.2.17}$$

The Boolean expression for the top event is:

$$TOP = G1 \cdot G2. \tag{2.2.18}$$

Then, substituting the Boolean expressions, for $G1$ and $G2$ into the TOP expression gives:

$$TOP = (X1 + X2) \cdot (X2 + X3). \tag{2.2.19}$$

Applying the Distributive law to the right part of the equation gives:

$$TOP = X2 + X1 \cdot X3. \tag{2.2.20}$$

This is the minimal cut set expression for the top event, in the form of Equation (2.2.1). The minimal cut sets are therefore $\{X2\}$, and $\{X1, X3\}$, which means if $X2$ fails or both $X1$ and $X3$ fail, then the top event will occur and the system will fail.

In practice, fault trees can contain large numbers of basic events and massive minimal cut sets. It will take intensive computation and extensive memory space to minimise and store the logic expressions for each gate. Some techniques have been applied to surmount this challenge, such as culling [32] in which some cut sets of high orders, say 4^{th} and above, are ignored or deleted during the calculation process. This technique is acceptable since cut sets with high orders tend to have low probability of occurrence and thus do not make significant contribution to the top event probability. However, this reduces the number of minimal cut sets that are found and will affect the accuracy of the following quantitative analysis.

2.2.4 Quantitative Analysis

The quantitative analysis in this research focuses on the use of basic event probability to calculate exact or approximated top event probability.

2.2.4.1 Inclusion-Exclusion Expansion

The top event probability is given by the *inclusion-exclusion expansion* as:

$$\begin{aligned} P(Top) =& P(\bigcup_{i=1}^{n} C_i) \\ =& \sum_{i=1}^{n} P(C_i) - \sum_{i=2}^{n}\sum_{j=1}^{i-1} P(C_i \cap C_j) + \dots \\ &+ (-1)^{n-1} P(C_1 \cap C_2 \cap \dots \cap C_n)\,, \end{aligned} \tag{2.2.21}$$

where C_i, $i = 1, 2, \dots, n$, and $P(C_i)$ is the probability of the i^{th} minimal cut set.

2.2.4.2 Example

Consider again the example fault tree shown in Figure 2.2.1, which has minimal cut sets $\{X2\}$ and $\{X1, X3\}$. Using the inclusion-exclusion expansion, Equation (2.2.21), the top event probability is given by:

$$\begin{aligned} P(Top) =& P(X2 + X1 \cdot X3) \\ =& P(X2) + P(X1 \cdot X3) - P(X2) \cdot P(X1 \cdot X3) \\ =& P(X2) + P(X1) \cdot P(X3) - P(X2) \cdot P(X1) \cdot P(X3). \end{aligned} \tag{2.2.22}$$

However, it will be a huge project to evaluate each term in the expression if there are many minimal cut sets, which is a normal situation in practical, industrial systems and even the fastest modern computers are incapable of conducting such complex calculations within a short time. Therefore, approximation methods are often used.

2.2.4.3 Approximation of Top Event Probability

The inclusion-exclusion expansion involves alternately adding odd-numbered terms and subtracting even-numbered terms, with each term being successively less numerically significant than the last [2].

Therefore, the first two terms of the inclusion-exclusion expansion provide a lower bound and the first term provides an upper bound of the system unreliability.

$$\sum_{i=1}^{n} P(C_i) - \sum_{i=2}^{n}\sum_{j=1}^{i-1} P(C_i \cap C_j) \le P(T) \le \sum_{i=1}^{n} P(C_i). \tag{2.2.23}$$

The upper bound provided by the first item is defined as *Rare Event Approximation*, and

is shown in Equation (2.2.24):

$$P_{Rare}(Top) = \sum_{i=1}^{n} P(C_i). \tag{2.2.24}$$

When the basic events are rare, i.e., with probabilities being almost 0, the orders of magnitude for the probabilities of the terms in the inclusion-exclusion formula rapidly decrease and converge to 0 exponentially. Therefore, terms with an order higher than two can be eliminated and this leads to a reasonably accurate approximation. When basic events are not rare, the rare event approximation result may not be accurate.

An alternative, more accurate approximation is the *Minimal Cut Set Upper Bound*:

$$P_{MCUB}(Top) = 1 - \prod_{i=1}^{n}(1 - P(C_i)). \tag{2.2.25}$$

The relations between the exact top event probability, the minimal cut set upper bound approximation and the rare event approximation are [2]:

$$P(Top) \leq P_{MCUB}(Top) \leq P_{Rare}(Top). \tag{2.2.26}$$

The minimal cut set upper bound approximation does not require the condition of rare events to achieve an accurate approximation result and Equation (2.2.26) also demonstrates P_{MCUB} is generally more accurate and closer to the exact probability than P_{Rare} is. Both of the exact calculations and approximations of the top event probability need the identification of minimal cut sets, which means quantitative analysis of fault trees can only be conducted after minimal cut sets are found. For practical, large systems, it can be impossible to identify all minimal cut sets because of time limitations and memory space restrictions, which in turn renders exact quantitative analysis impossible.

2.2.5 Summary

The fault tree represents the failure logic of a system in a visual format. Qualitative analysis gives the minimal cut sets, which are the smallest combinations of basic events that lead to the occurrence of the top event. The minimal cut sets are essential for the quantification of top event probability in the traditional fault tree analysis. Quantitative analysis gives exact or approximated numerical values of the top event probability. However, fault trees for practical system failures are often complex and need considerable time

and efforts to perform either qualitative or quantitative analysis, with it sometimes being impossible to generate any result. Binary Decision Diagrams, were introduced to address this problem and to improve the efficiency and accuracy of fault tree analysis for larger systems.

2.3 Binary Decision Diagrams

Binary Decision Diagrams (BDDs), were first introduced in [11] for the analysis of Boolean functions and then an algorithm was developed in [33] to convert fault trees directly to BDDs. Converting the fault tree into a BDD allows qualitative and quantitative analysis to be performed more efficiently.

2.3.1 The BDD Graph

A BDD is a directed acyclic graph, which means that all paths through the BDD are in one direction and no loops can exist. A BDD starts with a root node, which represents one of the fault tree basic events and has two branches leading from it. The 1-branch corresponds to the occurrence of the basic event, and the 0-branch corresponds to the non-occurrence of the basic event. A complete BDD consists of both terminal nodes and non-terminal nodes, connected by branches. Terminal nodes have a value of 1 or 0, which represent the occurrence and the non-occurrence of top event respectively. Non-terminal nodes represent basic events, which either occur or do not, according to whether they are traversed along their 1-branch or their 0-branch. An example BDD is shown in Figure 2.3.1. Each BDD node (or vertex) contains a Boolean variable (representing a fault tree basic event) and is connected to two child nodes by a 1-branch and a 0-branch. Node F contains variable $X1$ and has node $F1$ connected to it by its 1-branch and node $F0$ connected by its 0-branch. The size of a BDD is measured by the number of distinct non-terminal nodes it contains. The size of the BDD in Figure 2.3.1 is 3, since it contains the non-terminal nodes F, $F1$ and $F0$.

2.3.2 Fault Tree to BDD Conversion

2.3.2.1 BDD Conversion Rules

Rauzy [33] introduced an algorithm to convert a fault tree into a BDD. The algorithm is based on Shannon's Decomposition: Let f be a Boolean function on a set of Boolean

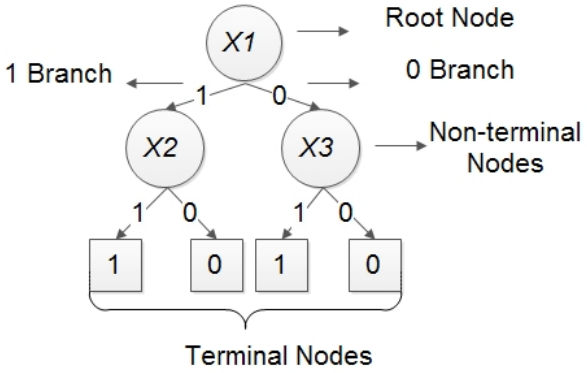

Figure 2.3.1: A BDD node illustration

variables X, and x be a Boolean variable in X, then

$$f = \{x = 1\} \cdot f_{x=1} + \{x = 0\} \cdot f_{x=0}, \tag{2.3.1}$$

where $f_{x=1}$ is the boolean function when $x = 1$ and $f_{x=0}$ is the boolean function when $x = 0$. To link Shannon's Decomposition with a BDD representation, an *if-then-else* (*ite*) structure is introduced as:

$$F = ite < x, F1, F0 >= x \cdot F1 + \bar{x} \cdot F0, \tag{2.3.2}$$

where x is an variable, $\bar{x}$ is the complement of x, $F1$ and $F0$ are both Boolean formulae in Shannon's form in which x doesn't occur. The expression presents that if x occurs, then consider $F1$, which corresponds to the 1-branch of the BDD node F, otherwise, consider $F0$, which corresponds to the 0-branch of F.

A variable ordering must be defined before a BDD is constructed, since the variables in a BDD must always appear in the same order when a BDD is traversed from its root node to any non-terminal node. The order of the variables affects the size of the BDD [11], and the effect of ordering schemes on BDD size is investigated later in Chapter 4.

Each basic event, A, in a fault tree is represented by an *ite* node $A = ite < A, 1, 0 >$ in a BDD. A BDD representing the same system failure logic as the fault tree is then constructed by combining *ite* nodes representing a gates's event and gate inputs according to the type of the gate and repeating this process until *ite* nodes have been developed for all events in the fault tree including the top event. Given a variable order, the construction of a BDD for a gate is as follows: suppose the two BDD nodes representing the gate inputs are $F = ite < x, F1, F0 >$ and $G = ite < y, G1, G0 >$ and suppose x is ordered before y

in the variable ordering scheme,i.e., $x \leq y$. Then

$$F \Diamond G = \begin{cases} ite < x, F1 \Diamond G, F0 \Diamond G >, & \text{if } x < y\,, \\ ite < x, F1 \Diamond G1, F0 \Diamond G0 >, & \text{if } x = y, \end{cases} \tag{2.3.3}$$

where $\Diamond$ is the Boolean operator, $AND(+)$ or $OR(\cdot)$, of the logic gate in the fault tree. Also, it is evident the following truth table applies for operations involving terminal 1 or 0 nodes:

$$\begin{aligned} 1 + G &= 1\,, \\ 1 \cdot G &= G\,, \\ 0 + G &= G\,, \\ 0 \cdot G &= 0. \end{aligned} \tag{2.3.4}$$

If both the 1-branch and 0-branch of the newly-constructed BDD node, $F \Diamond G$, connect to the same node, H, then $F \Diamond G = H$. This is because the state of the variable of $F \Diamond G$, x, does not affect the subsequent logic and therefore x can be eliminated from the BDD.

BDDs compactly encode fomulae in Shannon's form by means of subtree sharing. In other words, the efficiency of the BDD analysis derives from the way in which the *ite* nodes are stored. When a new node, $F = ite < x, F1, F0 >$, where x is a variable and F and G are two addresses in an *ite* node table, is calculated, the *ite* node table is consulted and F is stored only if it is not yet stored in the *ite* node table.

2.3.2.2 Example

Consider the fault tree in Figure 2.2.1 as an example. Given the ordering of $X2 < X1 < X3$, the *ite* structure of top event can be calculated as follows:

$$\begin{aligned} G1 &= ite < X2, 1, 0 > + ite < X1, 1, 0 > = ite < X2, 1, ite < X1, 1, 0 >> \,, \\ G2 &= ite < X2, 1, 0 > + ite < X3, 1, 0 > = ite < X2, 1, ite < X3, 1, 0 >> \,, \end{aligned} \tag{2.3.5}$$

$$
\begin{aligned}
TOP =& G1 \cdot G2 \\
=& ite < X2, 1, ite < X1, 1, 0 >> \cdot ite < X2, 1, ite < X3, 1, 0 >> \\
=& ite < X2, 1, ite < X1, 1, 0 > \cdot ite < X3, 1, 0 >> \\
=& ite < X2, 1, ite < X1, ite < X3, 1, 0 >, 0 >> \ .
\end{aligned}
\tag{2.3.6}
$$

Thus, the BDD for the fault tree shown in Figure 2.2.1 has the *ite* structure given in Equation (2.3.6), which represents the BDD shown in Figure 2.3.2.

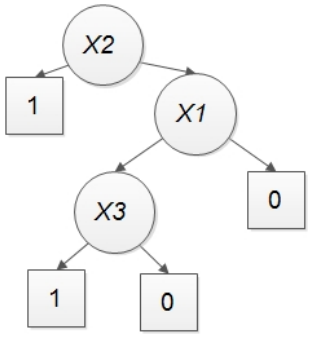

Figure 2.3.2: BDD for fault tree in Figure 2.2.1

2.3.3 Qualitative Analysis

The purpose of qualitative analysis is to identify the minimal cut sets, the minimal, necessary and sufficient conditions for the system failure mode modelled by the fault tree top event. The minimal cut sets are identified using a reduced BDD, which is a version of the original BDD that has been altered so that it encodes only the minimal cut sets. The reduced BDDs are obtained by applying a *WITHOUT* operator to the original BDD [33].

2.3.3.1 Obtaining Reduced-BDDs Containing Minimal Cut Sets

To illustrate the quantitative analysis of a BDD, consider the BDD given in Figure 2.3.3, which has the *ite* structure in Equation (2.3.7):

$$
F = ite < A, ite < B, 1, ite < C, 1, 0 >>, ite < C, 1, 0 >> \ . \tag{2.3.7}
$$

Each path through the BDD from the root node to a terminal 1 node gives a set of basic event conditions that result in the occurrence of the top event modelled by the BDD. Considering only the nodes that are traversed on their 1-branches gives a cut set relating

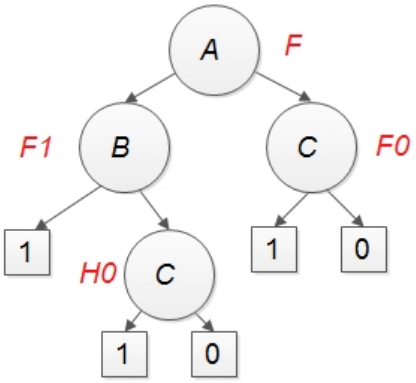

Figure 2.3.3: Example BDD

to each path.

For example, the basic event conditions relating to each of the paths ending at a terminal 1 node of the BDD in Figure 2.3.3 are:

$$\{A, B\}, \{A, \bar{B}, C\}, \{\bar{A}, C\}. \tag{2.3.8}$$

Taking only the elements of these paths that correspond to basic event occurrences gives three cut sets:

$$\{A, B\}, \{A, C\}, \{C\}. \tag{2.3.9}$$

Clearly, they are not minimal cut sets because set $\{A, C\}$ is redundant. This demonstrates the fact that a BDD does not directly encode the minimal cut sets of a fault tree, even though it encodes the system failure logic representing the top event.

Rauzy presented an approach to obtain the reduced BDD, which encodes only the minimal cut sets [33]. In order to obtain the minimal cut sets of the fault tree, the BDD, F, encoding the fault tree is first obtained, and then used to achieve a reduced BDD, F_{min}, based on F, such that the set of paths from the root of F_{min} to terminal 1 nodes defines exactly the minimal cut sets.

The algorithm is based on the theory: If $F = ite < x, G, H >$, the solutions of G and H are $sol_{min}(G)$ and $sol_{min}(H)$ respectively, then the solution of F, $sol_{min}(F)$, is given by:

$$sol_{min}(F) = \{\sigma | (\sigma = \delta \cup \{x\}) \cap (\delta \in sol_{min}(G)) \cap (\delta(H) = 0)\} \cup Sol_{min}(H). \tag{2.3.10}$$

The equation presents a recursive algorithm to compute the reduced BDD, F_{min}, which encodes the minimal solution of F, since it consists of computing G_{min} and H_{min}, which encode the minimal solutions of G and H, and removing all solution paths of H_{min} from G_{min}. Finally, combine the two obtained BDDs with x to obtain F_{min}.

The minimal solution of F, $sol_{min}(F)$ is achieved by recursive calculation of:

$$sol_{min}(F) = ite < x, sol_{min}(G)/H, sol_{min}(H) > . \tag{2.3.11}$$

A new operator, /, called the $WITHOUT$ operator appears in Equation (2.3.11) and is defined as follows: Let $F = ite < x, F1, F0 >$ and $G = ite < y, G1, G0 >$, then

1. If $x < y$,

$$F/G = ite < x, F1/G, F0/G > . \tag{2.3.12}$$

2. If $x > y$,

$$F/G = F/G0. \tag{2.3.13}$$

3. If $x = y$,

$$F/G = ite < x, F1/(G1 \text{ or } G0), F0/G0 > . \tag{2.3.14}$$

- The term $F1/(G1$ or $G0)$ in the last case means that each cut set in $F1$ is tested and deleted if it is a subset of a cut set in $G1$ or $G0$. It can be simplified to $F1/G0$ when the operation is applied to conventional BDDs.

Particularly, $sol_{min}(1) = 1$ and $sol_{min}(0) = 0$. The truth table for the without operator is shown as:

$$\begin{aligned} &0/G = 0. \\ &F/1 = 0. \\ &F/0 = F. \\ &1/G = 1 \end{aligned} \tag{2.3.15}$$

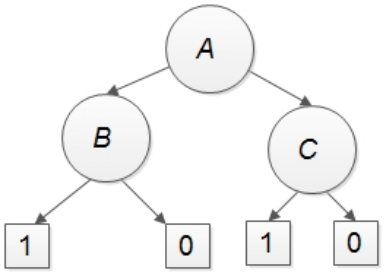

Figure 2.3.4: The reduced BDD obtained from the example BDD given in Figure 2.3.3

2.3.3.2 Example

The manipulation of the / operator is illustrated by applying it on the BDD shown in Figure 2.3.3, the minimal solutions of the root node F, $sol_{min}(F)$ are calculated using the following process:

$$\begin{aligned}
sol_{min}(H0) =& sol_{min}(F0) = ite < C, sol_{min}(1)/0, sol_{min}(0) >= ite < C, 1, 0 > \\
sol_{min}(F1) =& ite < B, sol_{min}(1)/H0, min(H0) > \\
=& ite < B, 1, min(H0) > \\
sol_{min}(F) =& ite < A, sol_{min}(F1)/F0, sol_{min}(F0) > \\
=& ite < A, ite < B, 1, ite < C, 1, 0 >> /ite < C, 1, 0 >, \\
& ite < C, 1, 0 >> \\
=& ite < A, ite < B, 1, ite < C, 0, 0 >>, ite < C, 1, 0 >> \\
=& ite < A, ite < B, 1, 0 >, ite < C, 1, 0 >>
\end{aligned} \tag{2.3.16}$$

The reduced BDD encoding the minimal solutions of F, $sol_{min}(F)$, is shown in Figure 2.3.4. The paths through this BDD yield the following cut sets:

$$\{A, B\}, \{C\}, \tag{2.3.17}$$

which are minimal.

The *WITHOUT* operator changes the logic structure of the original BDD (after reduction, Figure 2.3.3 becomes Figure 2.3.4, and therefore, the reduced BDD no longer encodes the failure logic of the original fault tree, quantifying which will lead to inaccurate top event probability calculation). The reduced BDD should therefore only be used to obtain the minimal cut sets.

2.3.4 Quantitative Analysis

As presented in Section 2.2.4, the quantitative analysis of a fault tree needs the identification of minimal cut sets, which are then used in the inclusion-exclusion expansion to evaluate the exact top event probability or in the rare event approximation or minimal cut set upper bound to evaluate an approximation. However, for exact evaluations, the calculation can be complex when the number of minimal cut sets is large, as $2^n - 1$ terms must be calculated to compute the sum of probabilities of the minimal cut sets. Approximations can reduce quantification time but lose accuracy. When using BDDs, the quantification can be carried out directly on the converted BDDs and the calculation complexity is linearly proportional to the size of the BDDs and does not require calculation of minimal cut sets. Quantification of the BDD yields an exact result and is a more efficient process than the previously-presented fault tree quantification.

2.3.4.1 Top Event Probability

The probability of the root node of a (non-reduced) BDD is the top event probability of the equivalent fault tree. It is expressed as the sum of the probabilities of the disjoint paths through the BDD. According to Shannon's Decomposition [33], if f is a Boolean function and x is a variable in f, then the probability of f is expressed by:

$$P(f) = P(x = 1) \cdot P(f_{x=1}) + P(x = 0) \cdot P(f_{x=0}). \tag{2.3.18}$$

Therefore, the probability of BDD node $F = ite < x, F1, F0 >$ is evaluated as:

$$P(F) = p(x) \cdot P(F1) + (1 - p(x)) \cdot P(F0), \tag{2.3.19}$$

where $p(x)$ is the failure probability of event x, $P(F)$ is the probability of the child node on the 1-branch, and $P(G)$ is the probability of the child node on the 0-branch. The calculation is performed recursively until terminal 1 and 0 nodes are reached, whose probabilities are 1 and 0 respectively.

The complexity of the calculation is linearly proportional to the size of the BDD and the calculated BDD node probability is cached in case the BDD node is a child node of more than one BDD node and the probability of it can then be used directly to avoid repeated computation.

2.3.4.2 Example

Consider the BDD in Figure 2.3.3, the probability of root node F is calculated as follows:

$$\begin{aligned} P(H0) &= P(F0) = p(C)\,, \\ P(F1) &= p(B) + (1 - p(B)) \cdot P(H0) = p(B) + (1 - p(B)) \cdot p(C)\,, \\ P(F) &= p(A) \cdot p(F1) + (1 - p(A)) \cdot P(F0) \\ &= p(A) \cdot (p(B) + (1 - p(B) \cdot p(C))) + (1 - p(A)) \cdot p(C)\,. \end{aligned} \tag{2.3.20}$$

The quantitative analysis thus must be applied on the original BDD (other than the reduced BDD). The root node probability of the reduced BDD is not equal to the top event probability of the original fault tree, since the reduced BDD algorithm changes the structure of the original BDD so that it no longer encodes the logic in the fault tree.

2.3.5 Summary

Fault trees for practical system failures are often complex and need considerable time and effort to perform either qualitative or quantitative analysis using the traditional method described in Section 2.2, with it sometimes being impossible to generate result. Therefore, BDDs were introduced to address this problem and to improve the efficiency and accuracy of fault tree analysis for large systems. By converting a fault tree to a BDD and quantifying the probability of the root node of the BDD, the exact top event probability of the equivalent fault tree is obtained in a more efficient way, since the way that BDD nodes are stored automatically applies sub-node sharing to avoid repeated work and the complexity of the probability calculation of the root node is linearly proportional to the size of the BDD. Therefore, better efficiency and accuracy can be obtained by employing the BDD model to analyse fault trees for practical, large systems. However, the BDD model described in this section is for standard fault trees with a single phase and single failure mode components. For systems where dependencies exist due to phases and components which can fail in more than one modes, BDD models must be adapted in order to address those dependencies and achieve accurate analysis results.

2.4 Phased Mission Systems (PMS)

Many engineering systems are capable of performing different tasks and these tasks must often be performed as phased missions, where the tasks are performed in sequence in order to achieve a specific objective. The periods in which each of these successive tasks takes place are known as *phases* and the sequence of phases that are completed in order to achieve an objective are known as a *mission*. For each phase of the mission, the system performs a specific task, each of which has different functional requirements and therefore requires a different system configuration and has its own failure criteria. A system may be capable of achieving a variety of objectives by varying the specific tasks it carries out along with their order and duration. A mission configuration is defined according to the tasks that must be completed, the time duration of each task and the sequence of the tasks.

To finish a phased mission successfully, each individual phase of that mission must be completed by the system without failure. In the case of occurrence of internal failures or external threats, the system must make decisions on whether to reconfigure itself or conduct alternative missions in order to finish the mission and to avoid catastrophic consequences. Within a mission, a component of the system can fail in any phase, and potentially leads to the system fails to complete that phase. In addition, a component's failure with no immediate effect to current phase can also lead to a system failure in later phases. A simple example would be the failure of the break system occurred during the steadily moving phase of a vehicle will not lead to any system failure until it tries to decelerate.

The systems that perform these phased missions are named PMS. An example of such a system is an aircraft, which is required to perform a flight mission consisting of 9 different tasks: starting up, taxiing out, taking off, climbing up to a cruising altitude, cruising, descending, landing, taxiing in and shutting down. During a flight, all of the tasks are completed in sequence with different time durations for each phase, as illustrated in Figure 2.4.1. The aircraft needs to finish all phases successfully to complete the flight.

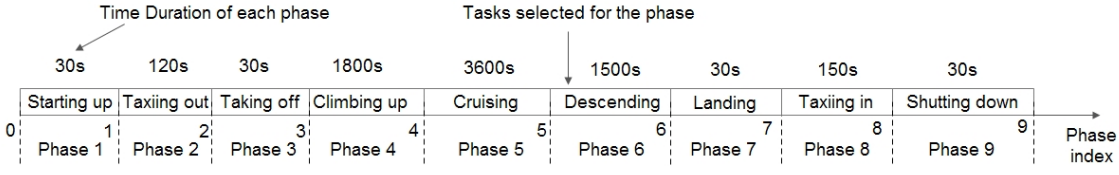

Figure 2.4.1: An example mission configuration for an aircraft flight

There are two categories of PMS: repairable systems where failed components may be restored to their working states during the mission, such as a ship with maintenance staff on board, where repair can be performed while the mission is underway; non-repairable systems where once a component has failed, it will remain in the failed state for the rest of the mission, for example, a missile which once launched, cannot be repaired. The phased mission systems considered in this research are non-repairable.

Many engineering systems contain components with multiple failure modes, which are mutually exclusive. For example, a switch in an engine system could fail to open or fail to close. In a non-repairable system, if the switch fails to open in a phase, then it will stay in the failure state in all the following phases and it will not be able to fail to close. This multiple failure mode character of the PMS is also taken into account during the course of this research.

2.5 Reliability Analysis of PMS

2.5.1 Introduction

Recall the reliability of a phased mission is defined as the probability of all the phases are completed without failure, while the unreliability is defined as the probability of any phase in the mission fails. In general, the reliability analysis of PMS is much more complicated due to the dependencies of components across phases. Various methods and tools have been developed to address this issue. Some of them [40][41][27] are based on Markov models, and construct Markov chains to represent the behaviour of both repairable and non-repairable systems, accounting for the dependencies arising from phases and multiple failure modes of one component. Although they can deal with dependencies, these models are faced with a state explosion problem in the case of too many components in a system. Other research based on combinatorial models, like fault tree analysis [46][44], assumes that the states of all components in a PMS are independent, which can simplify the analysis but is not accurate, since dependencies can arise across phases and multiple failure mode components.

Fault tree techniques are suitable when modelling non-repairable PMS, as those considered in this research, since the absence of repair processes facilitates the use of such techniques that require independence. However, the inclusion of multiple failure mode components introduces dependencies that make the analysis significantly more complex

than that of systems consisting of a single phase and single failure mode components. Despite the simple analysis it offers, the fault tree analysis can still not provide satisfying results for large systems within reasonable time frames [2]. This leads to the adoption of BDD models, which can greatly improve the accuracy and efficiency of fault tree analysis. The BDD model described in Section 2.3 is for the analysis of standard fault trees with single phases and single failure mode components and has been further developed for the analysis of PMS in later research, which will be detailed in Section 2.5.3.

This section first of all introduces how to model the failure of a phased mission and the failure of a phase conditional on the success of all previous phases using a fault tree representation. It is then followed by a review of BDD models for reliability analysis of PMS. Then, the BDD models for multiple platform PMS, where a number of systems work collaboratively to achieve a common mission objective, are reviewed.

2.5.2 Modelling Failures of PMS Using Fault Trees

2.5.2.1 Fault Tree Representation without NOT gates

The reliability analysis of PMS using fault trees was first introduced in [46], in which a method that transforms a multiple phase mission into a single phase mission in order to calculate the exact mission unreliability is developed. The failure logic of each phase is represented by a phase failure fault tree where the probability of failure of each component conditional on it having functioned until the start of current phase is calculated using its failure probability density function.

The logic expression for the failure of phase i is denoted as F_i and is represented by a phase failure fault tree. The logic expression for mission failure is denoted as F_{miss} and is represented by an OR gate with input (intermediate) events F_i, where $i = 1, 2 \cdots n$ and n is the total number of phases in the mission. The mission failure logic is given by Equation (2.5.1).

$$F_{miss} = F_1 + F_2 + \cdots + F_n. \tag{2.5.1}$$

In [46], the researchers attempted to evaluate the mission reliability by calculating the reliability of each phase individually using standard fault tree analysis and then multiplying the individual phase reliability together. However, the evaluation attempt does not generate the correct results, since the basic events in different phases are not always

independent, except in the case where none of the phases relies on common components. The inaccuracy resulting from the independency assumption can be significant.

A phase index subscript notation was introduced in [19] to represent the failure of a component within a certain time duration. Using this notation, the failure of component A between the end of phase m and the end of phase n is denoted as A_{mn} and particularly, the failure of component A between the start of the mission and the end of phase i is denoted as A_{0i}. This phase index notation will be used throughout this thesis to represent in which phases the failure of a component has occurred.

In order to obtain the exact mission reliability, a component failure event in the fault tree that represents failure of phase i (F_i), is replaced by an OR gate with input basic events standing for the failure of the component within a specific phase j, i.e, the component fails within phase j conditional on it having functioned in all the previous $j-1$ phases, for each phase j up to the end of current phase i, and is formulated as $A_{01} + A_{12} + \cdots + A_{i-1i}$, which can be further simplified to A_{0i} using the phase index subscription notion (since $A_{0i} = A_{01} + A_{12} + \cdots + A_{i-1i}$).

The mission unreliability, denoted as Q_{miss}, is obtained by calculating the top event probability of the fault tree representing F_{miss}:

$$Q_{miss} = P(F_{miss}) = P(F_1 + F_2 + \cdots + F_n). \tag{2.5.2}$$

The phased mission analysis technique in [46] was developed merely to calculate the mission unreliability or the probability of $F_1 + F_2 + \cdots + F_i$, representing a system failure having occurred at some point between the start of the mission and the end of phase m. The technique was unable to calculate the probability of a phase failure conditional on the system having successfully completed all the previous phases.

The logic expression for the conditional failure of phase i, i.e, a system failure has occurred in phase i conditional on the success of the previous $i-1$ phases, is denoted as ph_i and its probability is called the conditional unreliability of phase i and is denoted as $P(ph_i)$. A formula was derived in [29][3] to obtain $P(ph_i)$ using the probability of $F_1 + F_2 + \cdots + F_i$, $i = 2, 3, \cdots, n$, and is given in Equation (2.5.3).

$$\begin{aligned} P(ph_1) &= P(F_1)\,, \\ P(ph_i) &= P(F_1 + F_2 + \cdots + F_i) - P(F_1 + F_2 + \cdots + F_{i-1}). \end{aligned} \tag{2.5.3}$$

2.5.2.2 Fault Trees Representation with NOT gates

In [42], the researchers demonstrates the failure of phase i conditional on the success of the previous $i-1$ phases, can be represented directly by an AND logic of the failure of phase i and the success of all the previous $i-1$ phases. Researchers in [6] represent the logic expression using fault trees. The developed fault tree representation enables the conditional unreliability of each mission phase to be determined in addition to the whole mission unreliability. For any phase, the method combines the fault tree representing the success of all previous phases with the fault tree representing the failure of the current phase to allow both qualitative and quantitative analysis.

The success of phase i is represented by a NOT gate with the failure of phase i, F_i, as the only input, and is denoted as $\overline{F_i}$. The conditional failure of phase i is represented by an AND gate with input events representing the success of all phases j, $j = 1, \cdots, i-1$, $\overline{F_1}, \overline{F_2}, \overline{F_{i-1}}$, and the failure of phase i, F_i, which is represented by Equation (2.5.4):

$$\begin{aligned} ph_1 &= F1\,, \\ ph_2 &= \overline{F1} \cdot F2\,, \\ &\vdots \\ ph_i &= \overline{F1} \cdot \overline{F2} \cdots \overline{Fi-1} \cdot Fi. \end{aligned} \tag{2.5.4}$$

For any phase after the first phase, the incorporation of the success of the previous phases means that the fault tree will not simply consist of AND and OR gates. NOT logic will be required to represent the success of previous phases.

The unreliability of the whole mission is calculated by the sum of the conditional unreliabilities of phase 1 to phase n:

$$Q_{miss} = \sum_{i=1}^{n} P(ph_i). \tag{2.5.5}$$

2.5.2.3 Example

Consider for instance a system that can perform two tasks and the fault tree for failure of each task is shown in Figure 2.5.1. The system is required to perform a two phase mission with task 1 performed in phase 1 and task 2 performed in phase 2. Using fault trees without NOT gates, the failure of this mission is shown in Figure 2.5.2, where all basic events are associated with phase index subscription notion described in Section 2.5.2.1.

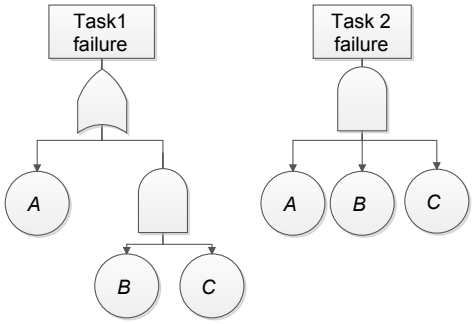

Figure 2.5.1: Fault trees representing failures of task 1 and task 2 of the example system

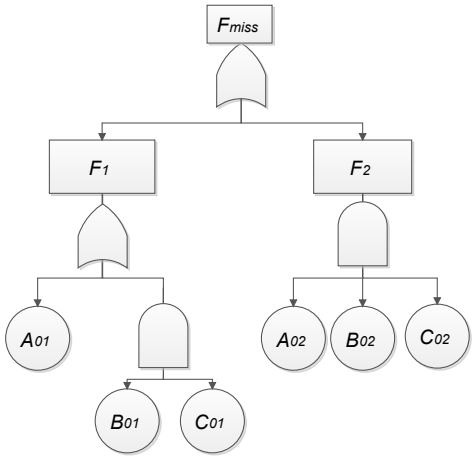

Figure 2.5.2: Mission failure represented by fault trees without NOT gates

Using fault trees with NOT gates, the (conditional) failure of phase 1 and the conditional failure of phase 2 are modelled in Figure 2.5.3.

The fault trees shown in Figure 2.5.2 and Figure 2.5.3 can then be used to quantify the conditional unreliability of the mission phases and the entire mission unreliability.

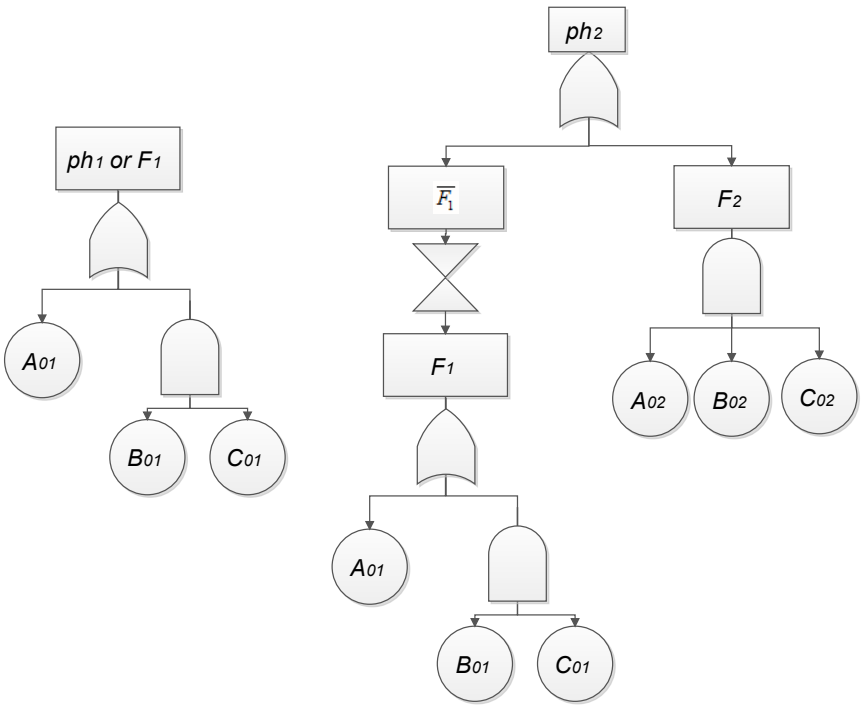

Figure 2.5.3: Conditional failure of phase 1 and phase 2 represented by fault trees with NOT gates

2.5.3 Using BDD Models to Analyse PMS

In the real world, PMS can be highly complex and the phased mission analysis for these types of system is computationally intensive. Compared with the traditional approach to fault tree analysis, which requires derivation of the minimal cut sets for use in the inclusion-exclusion expansion, BDDs are often seen as an efficient method of analysis for PMS. As presented in Section 2.3, BDD analysis requires independence between variables (corresponding to basic event independencies in fault tree analysis). In the general case, dependencies between variables can arise when basic events relate to different failure modes of the same component, while for PMS, as described in Section 2.4, dependencies arise due to the contribution of events to failure in different mission phases.

To deal with the dependencies within BDD analysis, two approaches have been developed and these approaches are reviewed in the following sections. One approach, as shown in [31][38][5], is to connect the BDDs together to represent the logic of conditional phase failure in each case, with dependencies being dealt with during the quantification process. For this approach, the BDD connection is fast but the quantification process is very time-consuming. In this case, a global variable ordering is not required and the variable ordering can differ for the BDDs of each phase. The other approach, investigated in [45][29][43][36], is to build the BDDs that account for these dependencies directly. The BDD construction process is relatively slow compared with the first approach, but once the BDDs have been constructed, the quantification process is relatively faster when compared to the first approach. In this case, a global variable ordering is necessary for all the phases in the mission.

2.5.3.1 Models not Considering Dependencies During BDD Construction

2.5.3.1.1 Introduction In order to allow the quantification of conditional phase unreliability and mission unreliability in a shortest time during a mission, researchers in [31][38][5] have presented a BDD-based method for phased mission analysis where dependencies exist between events relating to the same components but failing in different failure modes and those dependencies are not considered when constructing the BDDs that represent conditional phase unreliability. No global variable ordering is needed and thus BDDs representing functional tasks of the PMSs can be constructed using separate variable ordering lists, which allows the size of each BDD to be minimised.

Based on this, an online-offline strategyhas been proposed, which makes use of offline

and online computation in order to maximise the efficiency of analysis that is performed to support the decision making process during a mission, which will be discussed in Section 2.6. The main aim of the strategy is to carry out as much of the calculation as possible before the mission begins (offline) and to hence reduce the amount of computation that needs to be performed once the mission is underway (online). The online-offline strategy is illustrated in Figure 2.5.4.

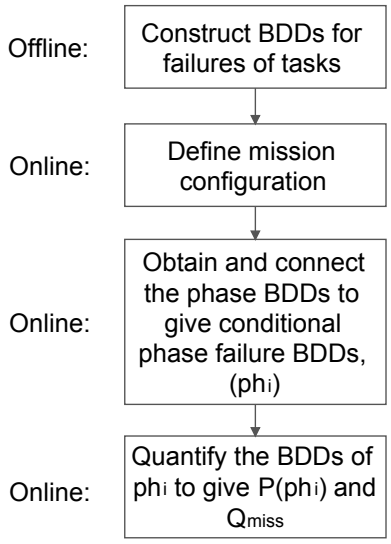

Figure 2.5.4: An online-offline strategy that is proposed to reduce the time of computation that needs to be performed when the mission is underway

2.5.3.1.2 BDD Construction In this approach to PMS analysis using BDDs, the main BDD construction tasks are carried out offline, so that they do not directly offset the time taken to begin the quantitative analysis required during the mission. BDDs are built to represent the structure function of each task that the PMS may be required to perform during a mission. The BDD for each task is made as small as possible, by selecting the ordering scheme that minimises the size of the BDD. The BDDs for task failure fault trees are constructed using Equation (2.3.3), following the assumption that all variables in each task failure fault tree are independent of each other.

After the tasks that must be performed in order to complete a mission have been identified, it is possible to define the phases in the mission using these tasks. The appropriate task BDDs are selected and the variables in these BDDs are associated with phases indices representing the phase within which the variables can contribute to the failure to complete the tasks.

The BDDs are connected according to the logic given in Equation (2.5.4). The success of phase i, denoted as $\overline{F_i}$, is represented by a *dual BDD*, which is created by swapping the terminal 1 and 0 nodes of the BDD for F_i. The AND connections in Equation (2.5.4) are represented by connecting all terminal one nodes of the BDD representing the previous phases to the root node of the BDD representing the current phase.

The connection is straight-forward and requires no processing of variables since dependencies between the variables are considered during the quantification process. This direct connection saves a large amount of BDD construction time for each ph_i due to the fact that it allows each BDD to be treated as independent when the BDDs are connected.

2.5.3.1.3 Quantitative Analysis Due to the construction process, which is performed by simple connections of the BDDs without consideration of the dependencies that exist between variables, quantitative analysis must account for these dependencies in order to be accurate. This analysis is performed by considering the paths through the BDD from the root node to its terminal 1 nodes. A path is represented by an AND operation of successive variables starting from the root node, tracing through the whole BDD and ending at terminal 1 or 0 nodes. Passing through a node along its 1-branch corresponds to the occurrence of the variable related to that node, while passing through a node along its 0-branch corresponds to the non-occurrence of the variable.

Generally, paths through the BDDs representing the conditional failure of phase i, ph_i, will contain variables that relate to the same components but have different phase indices or failure modes, meaning that they are dependent. In order to allow quantification, simplification rules must be applied to remove the dependencies between variables and ensure that the variables on a path are in their simplest forms.

The simplification rules for solving the dependencies between events relating to the same component but failing in different phases are given by Equation (2.5.6) [19].

$$\begin{aligned} \overline{A_{0i}} &= A_{i\infty}\,, \\ A_{ij} &= 0 \quad (i > j)\,, \\ A_{i_1 j_1} \cdot A_{i_2 j_2} &= A_{min(i_1,i_2)max(j_1,j_2)}. \end{aligned} \tag{2.5.6}$$

The simplification rules for solving the dependencies between events relating to the same component but failing in different phases and failure modes are given by Equation

(2.5.7) [31]. For cases where the phase index $i > j$ and for any failure mode $m \neq n$:

$$\begin{aligned}
\overline{A_{0i}^m} &= A_{i\infty}^m \,, \\
A_{ij}^m &= 0 \,, \\
A_{i_1 j_1}^m \cdot A_{i_2 j_2}^m &= A_{max(i_1,i_2),min(j_1,j_2)}^m \,, \\
A_{i_1 j_1}^m \cdot A_{i_2 j_2}^n &= 0 \,, \\
\overline{A_{0i_1}^m} \cdot A_{t_{i_2} t_{j_2}}^n &= A_{i_2 j_2}^n .
\end{aligned} \tag{2.5.7}$$

The probability of a path terminating at a terminal 1 node is calculated by the product of the probabilities of all independent terms on the path, which requires simplification of the path to allow dependencies to be accounted for. Since the paths through the BDD are disjoint, the probability of conditional phase i failure, $P(ph_i)$, is calculated by the sum of the probabilities of all paths terminating at node 1:

$$P(ph_i) = \sum_{i:\text{path terminated at 1}} P(path_i). \tag{2.5.8}$$

The probabilities of variables relating to the same component in a path are calculated according to the component's failure probability density function if a single term is obtained after simplification and Equation (2.5.9) if one or more than one non-occurrence term is obtained.

$$P(\overline{A_{0i_1}^{m_1}} \cdot \overline{A_{0i_2}^{m_2}} \cdots \overline{A_{0i_k}^{m_k}}) = 1 - p(A_{0i_1}^{m_1}) - p(A_{0i_2}^{m_2}) - \cdots - p(A_{0i_k}^{m_k}). \tag{2.5.9}$$

The unreliability of the whole mission is given by adding the conditional phase unreliabilities. For a mission with n phases [6]:

$$Q_{miss} = \sum_{i=1}^{n} P(ph_i) \tag{2.5.10}$$

The steps of calculating the conditional unreliability of each phase and the mission unreliability using the BDD model which constructs BDDs without considering dependencies can be summarised as follows:

1. Choose the ordering scheme that leads to the smallest BDD size (which will be discussed in Chapter 4) and construct the BDD for each task failure fault tree using the BDD rules for fault trees of single phase systems in Equation (2.3.3).

2. Connect BDDs together according to Equation (2.5.4) to obtain the BDD representing the conditional failure of each phase i, ph_i.

3. Use Equation (2.5.6) to obtain the path sets that contribute to the occurrence of the top event of the fault tree standing for ph_i.

4. Use defined component failure density functions and Equation (2.5.9) to calculate the probability of each path that contributes to ph_i and use Equation (2.5.8) to obtain the conditional unreliability of phase i, p(ph_i).

5. Use Equation (2.5.10) to obtain the unreliability of the whole mission.

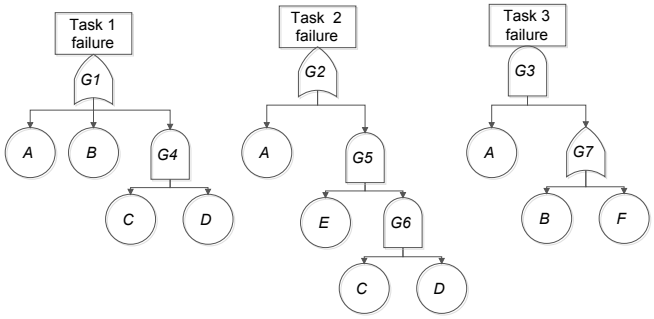

Figure 2.5.5: Fault trees of a system can perform 3 tasks. Each tree represents the failure of a single task.

2.5.3.1.4 Example Consider a system that can perform three tasks, with fault trees representing the failure to complete those tasks as shown in Figure 2.5.5.

The variables in the task fault trees are ordered as follows prior to BDD construction:

- Task 1: $A < B < C < D$.
- Task 2: $A < E < C < D$.
- Task 3: $A < B < F$.

The BDDs corresponding to the fault trees are shown in Figure 2.5.6. Suppose the system is required to perform a mission with three phases, which are task 1, task 3 and task 2, in sequence. The BDDs constructed for each phase are then represented in Figure 2.5.7 by associating start and end phase indices with the variables in the task BDDs. These indices consider the time period over which the variables can contribute to the current phase failure. The resultant BDDs can then be used to calculate the mission unreliability and conditional phase unreliability, using the connection method described above to represent

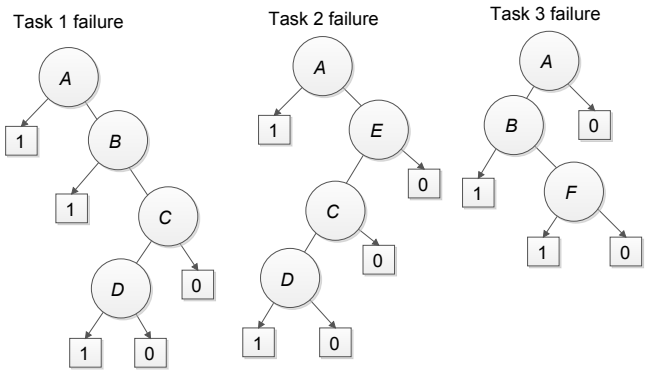

Figure 2.5.6: BDDs representing the failure of tasks the system capable of

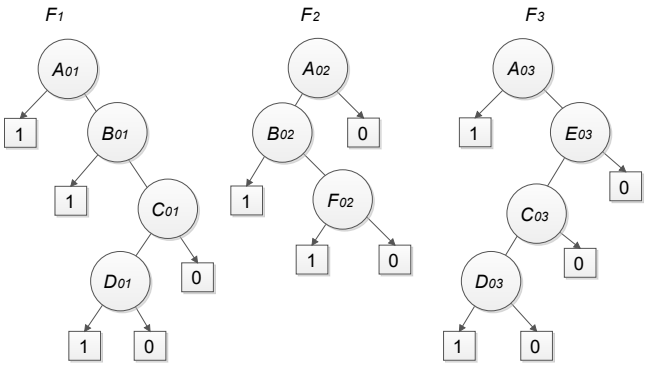

Figure 2.5.7: BDDs for F_i

conditional phase failure. ph_1 has the same structure as F_1, and ph_2 is represented by first swapping the terminal 1 and terminal 0 nodes of F_1 and then connecting the resultant terminal 1 nodes with the root node of F_2. ph_3 is represented by first swapping all terminal 1 and 0 nodes in ph_2 that belong to F_2, and then connecting the resultant terminal 1 nodes with the root node of F_3. The BDDs representing conditional failure in each phase are illustrated in Figure 2.5.8.

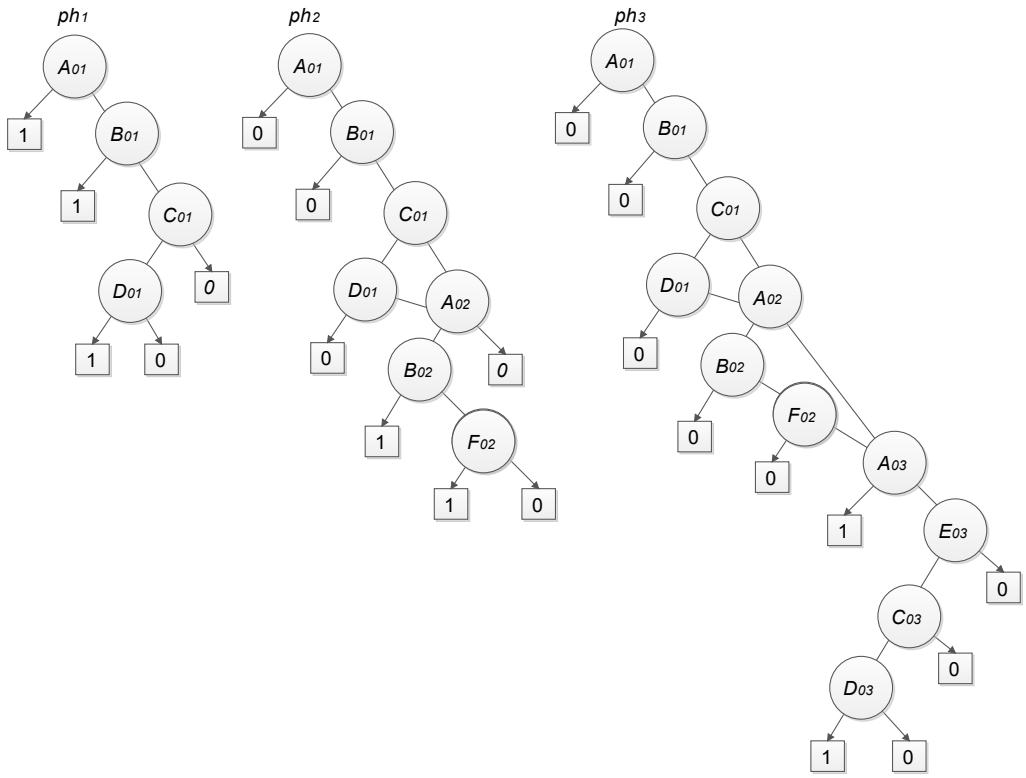

Figure 2.5.8: BDDs for ph_1, ph_2, ph_3

The component dependencies across the phases are addressed by accounting for the variable dependencies along each path before quantification. For ph_1, by tracing from the root node A_{01} to terminal 1 nodes, 3 disjoint paths are identified:

Path 1.1 A_{01} ,

Path 1.2 $\bar{A}_{01}B_{01}$,

Path 1.3 $\bar{A}_{01}\bar{B}_{01}C_{01}D_{01}$.

None of these paths needs further simplification, as all variables on each path are inde-

pendent. Using Equation (2.5.8), the conditional phase 1 unreliability is given by:

$$\begin{aligned} P(ph_1) =& P(F_1) = P(A_{01}) + P(\bar{A}_{01}B_{01}) + P(\bar{A}_{01}\bar{B}_{01}C_{01}D_{01}) \\ =& p(A_{01}) + [1-p(A_{01})]p(B_{01}) + [1-p(A_{01})][1-p(B_{01})]p(C_{01})p(D_{01}). \end{aligned} \tag{2.5.11}$$

For ph_2, the disjoint paths are:

Path 2.1 $\bar{A}_{01}\bar{B}_{01}C_{01}\bar{D}_{01}A_{02}B_{02}$,

Path 2.2 $\bar{A}_{01}\bar{B}_{01}\bar{C}_{01}A_{02}B_{02}$,

Path 2.3 $\bar{A}_{01}\bar{B}_{01}C_{01}\bar{D}_{01}A_{02}\bar{B}_{02}F_{02}$,

Path 2.4 $\bar{A}_{01}\bar{B}_{01}\bar{C}_{01}A_{02}\bar{B}_{02}F_{02}$.

Applying the simplification rules in Equation (2.5.7) gives the following expression for each path:

Path 2.1 $A_{12}B_{12}C_{01}\bar{D}_{01}$,

Path 2.2 $A_{12}B_{12}\bar{C}_{01}$,

Path 2.3 $A_{12}\bar{B}_{02}C_{01}\bar{D}_{01}F_{02}$,

Path 2.4 $A_{12}\bar{B}_{02}\bar{C}_{01}F_{02}$.

The conditional unreliability of phase 2, $P(ph_2)$, is given by:

$$\begin{aligned} P(ph_2) =& P(A_{12}B_{12}C_{01}\bar{D}_{01}) + P(A_{12}B_{12}\bar{C}_{01}) + P(A_{12}\bar{B}_{02}C_{01}\bar{D}_{01}F_{02}) + P(A_{12}\bar{B}_{02}\bar{C}_{01}F_{02}) \\ =& p(A_{12})p(B_{12})p(C_{01})[1-p(D_{01})] + p(A_{12})p(B_{12})[1-p(C_{01})] \\ &+ p(A_{12})[1-p(B_{02})]p(C_{01})[1-p(D_{01})]p(F_{02}) \\ &+ p(A_{12})[1-p(B_{02})][1-p(C_{01})]p(F_{02}). \end{aligned} \tag{2.5.12}$$

The BDD representing the failure of the system in phase 3 failure, conditional on the success of the previous two phases, ph_3, has 8 disjoint paths:

Path 3.1 $\bar{A}_{01}\bar{B}_{01}C_{01}\bar{D}_{01}A_{02}\bar{B}_{02}\bar{F}_{02}A_{03}$,

Path 3.2 $\bar{A}_{01}\bar{B}_{01}\bar{C}_{01}A_{02}\bar{B}_{02}\bar{F}_{02}A_{03}$,

Path 3.3 $\bar{A}_{01}\bar{B}_{01}C_{01}\bar{D}_{01}\bar{A}_{02}A_{03}$,

Path 3.4 $\bar{A}_{01}\bar{B}_{01}\bar{C}_{01}\bar{A}_{02}A_{03}$,

Path 3.5 $\bar{A}_{01}\bar{B}_{01}C_{01}\bar{D}_{01}A_{02}\bar{B}_{02}\bar{F}_{02}\bar{A}_{03}E_{03}C_{03}D_{03}$,

Path 3.6 $\bar{A}_{01}\bar{B}_{01}\bar{C}_{01}A_{02}\bar{B}_{02}\bar{F}_{02}\bar{A}_{03}E_{03}C_{03}D_{03}$,

Path 3.7 $\bar{A}_{01}\bar{B}_{01}C_{01}\bar{D}_{01}\bar{A}_{02}\bar{A}_{03}E_{03}C_{03}D_{03}$,

Path 3.8 $\bar{A}_{01}\bar{B}_{01}\bar{C}_{01}\bar{A}_{02}\bar{A}_{03}E_{03}C_{03}D_{03}$.

Applying the simplification rules in Equation (2.5.7) to get the following expressions for each path:

Path 3.1 $A_{12}\bar{B}_{02}C_{01}\bar{D}_{01}\bar{F}_{02}$,

Path 3.2 $A_{12}\bar{B}_{02}\bar{C}_{01}\bar{F}_{02}$,

Path 3.3 $A_{23}\bar{B}_{01}C_{01}\bar{D}_{01}$,

Path 3.4 $A_{23}\bar{B}_{01}\bar{C}_{01}$,

Path 3.5 0 ,

Path 3.6 0 ,

Path 3.7 $\bar{A}_{03}\bar{B}_{01}C_{01}D_{13}E_{03}$,

Path 3.8 $\bar{A}_{03}\bar{B}_{01}C_{13}D_{03}E_{03}$.

The conditional unreliability of phase 3 is then can be calculated by summing the probabilities of the 8 paths:

$$\begin{aligned}
P(ph_3) =& P(A_{12}\bar{B}_{02}C_{01}\bar{D}_{01}\bar{F}_{02}) + P(A_{12}\bar{B}_{02}\bar{C}_{01}\bar{F}_{02}) + P(A_{23}\bar{B}_{01}C_{01}\bar{D}_{01}) \\
&+ P(A_{23}\bar{B}_{01}\bar{C}_{01}) + P(\bar{A}_{03}\bar{B}_{01}C_{01}D_{13}E_{03}) + P(\bar{A}_{03}\bar{B}_{01}C_{13}D_{03}E_{03}) \\
=& p(A_{12})[1-p(B_{02})]p(C_{01})[1-p(D_{01})][1-p(F_{02})] \\
&+ p(A_{12})[1-p(B_{02})][1-p(C_{01})][1-p(F_{02})] \\
&+ p(A_{23})[1-p(B_{01})]p(C_{01})[1-p(D_{01})] + p(A_{23})[1-p(B_{01})[1-p(C_{01})] \\
&+ [1-p(A_{03})][1-p(B_{01})]p(C_{01})p(D_{13})p(E_{03}) \\
&+ [1-p(A_{03})][1-p(B_{01})]p(C_{13})p(D_{03})p(E_{03}).
\end{aligned} \tag{2.5.13}$$

The mission unreliability is then calculated by adding the conditional phase unreliability, given in Equation (2.5.11), Equation (2.5.12), and Equation (2.5.13):

$$Q_{miss} = P(ph_1) + P(ph_2) + P(ph_3). \tag{2.5.14}$$

2.5.3.2 Models Considering Dependencies During BDD Construction

2.5.3.2.1 Introduction In [45] [29] [3], the BDD analysis of PMSs with single failure mode components is investigated, where BDDs are constructed considering the dependencies between phases. For this approach, the BDD construction for conditional phase failure analysis takes the majority of the analysis time. Once the BDDs have been constructed, their quantification can be performed rapidly compared with the analysis method in Section 2.5.3.1. Since these methods of analysis require dependencies to be considered during construction of the BDDs, a global variable ordering must be arranged, which ensures consistency of ordering in all of the task BDDs.

Constructing the BDDs in this way ensures that dependencies to be considered between phases are accounted for in the final BDD structure, meaning that any dependencies that must be considered during quantification occur between variables in connected nodes unlike the situation for the methods considered in the Sections 2.5.3.1. This reduces the computational burden during quantification, and the constructed BDD can therefore be efficiently evaluated.

In [45], a BDD model for the analysis of PMS (PMS-BDD) was introduced and the work in [29] [3] improved the analysis efficiency of certain PMSs by taking modules (as discussed in Section 2.2) into account. All of these research considered PMS containing single failure mode components and research in [43][36] extended the BDD analysis to PMS containing multiple failure mode components, which will be detailed in Chapter 3. In this section, the BDD analysis proposed in [45] and the BDD analysis proposed in [3], which makes use of modularisation technique, are reviewed.

2.5.3.2.2 BDD construction When constructing the BDDs representing the conditions for mission failure, F_{miss}, it is necessary to take account of dependencies between variables that arise due to the fact that PMS are being studied. In order to do this, a phase algebra is used, which details how these dependencies are accounted for in PMS with single failure mode components [45].

To illustrate the use of the phase algebra, consider two variables, A_{0i} and A_{0j}, which

relate to two events representing the same component but contributing to the failure of different phases. A_{0i} means component A fails between the start of the mission and the end of phase i while A_{0j} means A fails between the start of the mission to end of phase j. $\overline{A_{0i}}$ is the complement of A_{0i}, meaning component A does not fail between the start of the mission and the end of phase i. The phase algebra dealing with the relation between the two events is given by Equation (2.5.15): (suppose $i < j$)

$$\begin{aligned}
&A_{0i} \cdot A_{0j} = A_{0i}\,, \\
&\overline{A_{0i}} \cdot \overline{A_{0j}} = \overline{A_{0j}}\,, \\
&\overline{A_{0i}} \cdot A_{0j} = A_{ij}\,, \\
&\overline{A_{0i}} + \overline{A_{0j}} = \overline{A_{0i}}\,, \\
&A_{0i} + A_{0j} = A_{0j}\,, \\
&\overline{A_{0i}} + A_{0j} = 1\,, \\
&A_{0i} \cdot \overline{A_{0j}} = 0.
\end{aligned} \tag{2.5.15}$$

A global variable ordering is required before construction of the BDDs representing mission failure, because events relating to the same components needs to be adjacent within the BDD structure so that Boolean manipulation can be performed during BDDs construction to address the dependencies between them. For systems with single failure modes, two levels of ordering must be considered: component level ordering and phase level ordering. Component level ordering requires variables to be ordered according to an ordering heuristic applied only to the components, while phase level dictates how variables relating to the same component are ordered. This is usually according to a forward phase index, where variables with lower ending phase index appear before those with a higher ending phase index, or backward phase index, where the situation is reversed. Component level ordering is given higher priority than phase level ordering, thus variables are ordered first according to the components to which they are related and then according to their phase, using forward or backward phase ordering.

To analyse the conditional phase unreliability and mission unreliability, Equation (2.5.3) and Equation (2.5.2) are used, where BDDs for $F_1 + \cdots + F_i$, $i = 2, 3, \cdots n$ must be constructed. In [45], researchers developed the forward phase dependent operator (PDO) and backward PDO according to the forward phase index ordering and backward phase index ordering to allow the combination of two BDD nodes using a logic operator which con-

sidered phase dependencies. Consider the operation of two nodes $F = ite < x, F1, F0 >$ and $G = ite < y, G1, G0 >$, where $x \leq y$. Equation (2.3.3) is used if x and y relate to different components or $x = y$, since this will be equivalent to the standard case. If x and y relate to the same component, the forward PDO is applied using Equation (2.5.16):

$$ite < x, F1, F0 > \diamond ite < y, G1, G0 >= ite < x, F1 \diamond G1, G0 \diamond G >, \tag{2.5.16}$$

because under forward phase ordering, the occurrence of x ($x = 1$) implies the occurrence of y ($y = 1$); the backward PDO is applied using Equation (2.5.17),

$$ite < x, F1, F0 > \diamond ite < y, G1, G0 >= ite < x, F1 \diamond G, F0 \diamond G0 >, \tag{2.5.17}$$

because under backward phase ordering, the non-occurrence of x ($x = 0$) implies the non-occurrence of y ($y = 0$).

The test that compares the two PDO in [45] shows that backward PDO generated a smaller BDD size compared with forward PDO, since variables relating to the same component but appearing in early phases can be automatically cancelled during the generation of a BDD without the need for additional computation. Consider for example, $A_{02} = 0$ which implies $A_{01} = 0$. If this is considered during the construction of BDDs using backward PDO, then A_{01} is eliminated from the BDD and thus the BDD will be in a more compact format. (This is stated to be true in general in [45], but this is seen not be the case in [36], a situation that is supported by the results of Chapter 3.)

2.5.3.2.3 Quantitative Analysis Quantitative analysis to find the probability of the root node requires consideration of the fact that connecting nodes within the BDD may contain dependent variables due to the fact that they relate to the same component. A recursive quantification method is proposed in [45] to quantify BDDs using backward phase ordering. The quantification method allows the probabilities to be cached for each BDD node, avoiding repeated calculations for nodes with multiple parents. In BDDs constructed using the backward PDO, the 0-branch always links to variables that belong to two different components. However, two different cases are possible for the two variables linked by the 1-branch: they can relate to different components or they can relate to the same component but have a different phase index. For a BDD node: $G = ite < x, G1, G0 >= x \cdot G1 + \bar{x} \cdot G0$, where $G1 = ite < y, H1, H0 >= y \cdot H1 + \bar{y} \cdot H0$, the

probability of G is calculated using Equation (2.5.18).

$$P(G) = \begin{cases} p(x) \cdot P(G1) + (1 - p(x)) \cdot P(G0)\,, & \text{if } x,\ y \text{ relate to different components,} \\ P(G1) + (1 - P(x)) \cdot (P(G0) - P(H0))\,, & \text{if } x,\ y \text{ relate to same component.} \end{cases} \tag{2.5.18}$$

The steps of calculating the conditional unreliability of each phase and the mission unreliability using the BDD model which constructs BDDs while considering dependencies are summarised as follows:

1. Construct BDDs for each fault tree representing $F_1 + \cdots + F_i$, for each $i = 2, 3, \cdots, n$ using Equation (2.5.17) and Equation (2.3.3) under a global variable ordering defined before construction.

2. Use Equation (2.5.18) to obtain the probability of the root node of the obtained BDD and use Equation (2.5.3) and Equation (2.5.2) to calculate the conditional unreliability of the phases and the whole mission.

2.5.3.2.4 Example Consider the same example system whose task failure fault trees are illustrated in Figure 2.5.5, with an identical mission to that described in Section 2.5.3.1.4. To use the BDD analysis represented in [45], a global variable ordering scheme is required before the task failure fault trees are converted into BDDs. Suppose the components are ordered: $A < B < C < D < E < F$. Consider that backward phase ordering is used to order variables across phases and correspondingly, the backward PDO is used to construct BDDs for the different phases. The variables relating to the different phases are thus ordered as follows:

$$A_{03} < A_{02} < A_{01} < B_{02} < B_{01} < C_{03} < C_{01} < D_{03} < D_{01} < E_{03} < F_{02}\,.$$

This variable ordering results in the BDDs shown in Figure 2.5.9 and the BDDs representing the OR combination of the failure conditions for the first three phases, F_1, $F_1 + F_2$, and $F_1 + F_2 + F_3$, are constructed using Equation (2.5.17). The resulting BDDs are shown in Figure 2.5.10, with each BDD node denoted with a red $N + number$ label.

The conditional phase unreliability is obtained using Equation (2.5.3) where the probability of $F_1 + F_2 + \cdots + F_i$ is the root node probability of the BDDs in Figure 2.5.10. First, the probabilities of F_1, $F_1 + F_2$ and $F_1 + F_2 + F_3$ are calculated using Equation

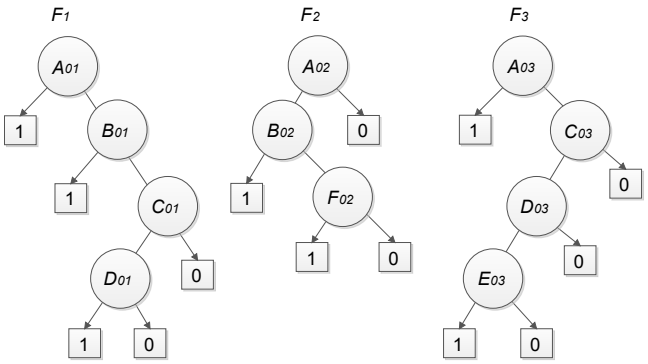

Figure 2.5.9: BDD for each F_i

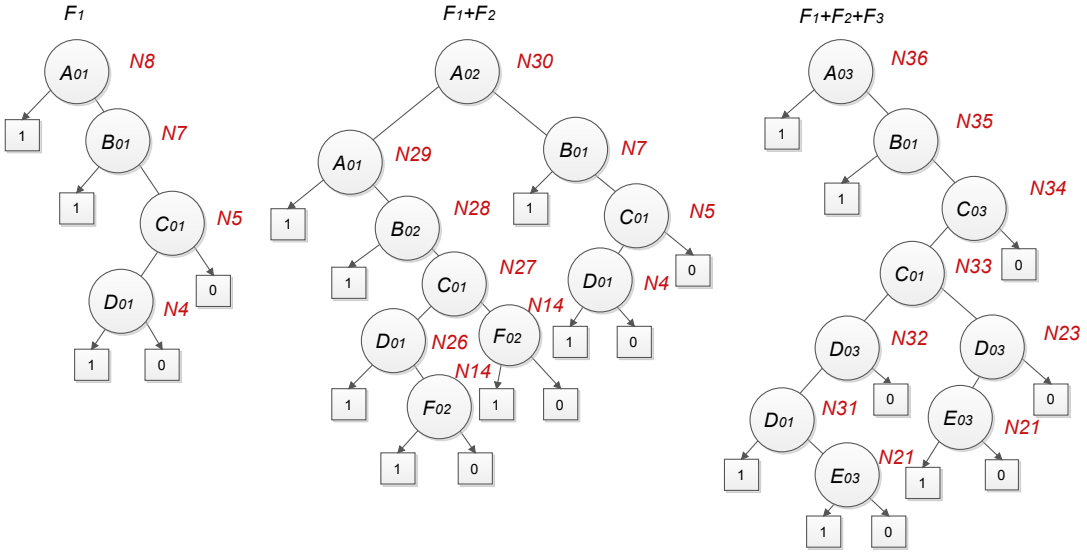

Figure 2.5.10: BDDs for F_1, $F_1 + F_2$ and $F_1 + F_2 + F_3$

(2.5.18), then the conditional unreliability of each phase, $P(ph_1)$, $P(ph_2)$ and $P(ph_3)$ can be calculated using Equation (2.5.3)

$$
\begin{aligned}
P(ph_1) =& P(F1) = P(N8) = p(A_{01}) \cdot 1 + [1 - p(A_{01})]P(N7) \\
=& p(A_{01}) + [1 - P(A_{01})][p(B_{01}) + [1 - p(B_{01})]p(C_{01})p(D_{01})] \,, \\
P(F1 + F2) =& P(N30) = P(N29) + [1 - p(A_{02})][P(N7) - P(N28)] \\
=& p(A_{01}) + [1 - p(A_{01})][p(B_{02}) + [1 - p(B_{02})][p(C_{01})[p(D_{01}) \\
& + [1 - p(D_{01})]p(F_{02})] + [1 - p(C_{01})]p(F_{02})]] \\
& + [1 - p(A_{02})][p(B_{01}) + [1 - p(B_{01})]p(C_{01})p(D_{01}) \\
& - [p(B_{02}) + [1 - p(B_{02})][p(C_{01})[p(D_{01}) \\
& + [1 - p(D_{01})]p(F_{02})] + [1 - p(C_{01})]p(F_{02})]]] \,, \\
P(ph_2) =& P(F1 + F2) - P(F1) \,, \\
Q_{miss} =& P(F1 + F2 + F3) = P(N36) = p(A_{03}) + [1 - p(A_{03})]P(N35) \\
=& p(A_{03}) + [1 - p(A_{03})][p(B_{01}) + [1 - p(B_{01})][p(C_{01})[p(D_{01}) + [1 - p(D_{01})]p(E_{03}) \\
& - p(E_{03})[1 - p(D_{03})]] + [1 - p(C_{01})]p(D_{03})p(E_{03}) - [1 - p(C_{03})]p(D_{03})p(E_{03})]] \,, \\
P(ph_3) =& P(F_1 + F_2 + F_3) - P(F_1 + F_2) \,.
\end{aligned}
$$

2.5.3.3 BDD Analysis with Modularisation

2.5.3.3.1 Introduction In [29][3], researchers pointed out that some functionalities of a system can affect almost all the phases in a mission. For example, the power supply for a system, or certain key aspects of functionality on an aircraft, like thrust, is required in each phase of a flight. The failure of these common subsystems may affect all mission phases. Since the failure of common subsystems can appear frequently throughout the phased mission fault trees, treating these common subsystem failures as a single complex event can allow a more efficient analysis. One way of taking advantage of these features is by identifying modules of the phased mission fault trees, since the modules can be analysed separately from the rest of the fault tree to simplify the analysis. Modules for a single phase fault tree can be detected using the linear-time algorithm described in Section 2.2.2. This algorithm is further extended in [3][29] to find modules for phased missions.

The efficiency of BDD analysis of PMS could be improved through applying such a modularisation technique and will bring benefits when the number of modules in a phased mission fault tree is large. The BDD models in [3][29] use the same BDD analysis method

as described in Section 2.5.3.2 while taking the modules of the PMS into consideration to determine the mission unreliability and conditional unreliability of mission phases as soon as the mission is defined. The model also adopts the online-offline strategy described in Section 2.5.3.1 in order to calcualte the unreliability in the shortest possible time.

2.5.3.3.2 Finding PMS Modules Since the analysis method uses the same BDD model as described in Section 2.5.3.2, a global variable ordering is required, which will remain the same no matter how the configuration of the system in the mission being performed changes. Using the BDD rules for single phase systems described in Section 2.3, the fault trees representing the failure of tasks that can be used to define any mission that a system can perform are converted into BDDs. To support the online calculations, a library is constructed which contains all the converted BDDs representing task failures together with all module information.

Besides the analysis required using the BDD model described in Section 2.5.3.2, this model needs additional analysis to obtain module information for the task fault trees and the whole mission after the mission has been defined. The additional steps to identify the module information are as follows:

1. Modules for each task failure fault tree are identified using the linear-time algorithm described in Section 2.2.2. During the analysis, the selected modules are replaced by module events.

2. A module hierarchy must be established since a BDD for a module can only be constructed if all of its constituent modules have been identified and constructed. When comparing the level of two modules, firstly, they must be checked to see whether they belong to the same phase. If not, the following processes, 2.(a) and 2.(b), are unnecessary. If they belong to the same phase, then denote the first gate as the *above gate* and the second gate as the *below gate*. The below gate is beneath the above gate if either of the following two conditions is satisfied:

   (a) The first visit step number of the above gate is less than the first visit step number of the below gate, and the second visit step number of the above gate is larger than the second visit step number of the below gate.

   (b) The first visit step number of the above gate is less than the third or more visit step number of the below gate, which is simultaneously less than the second visit step number of the above gate.

3. The contradiction tasks of the modules, defined as the task that once selected, results in the considered task fault tree module no longer being a module for the phased mission fault tree, must be identified.

   Before a mission is defined, the tasks that it comprises are unknown and an event under a module may not appear anywhere in a task fault tree but may appear somewhere in the phased mission fault tree, meaning that those modules that are mission modules and those that are not is unclear. A mission module can only be identified by comparing the contradiction tasks of the module to the tasks selected in the mission after the mission is defined. The contradiction tasks for a potential module are obtained by:

   (a) Listing all basic events that occur under the module.

   (b) Scanning the chosen task fault tree and listing all basic events in them.

   (c) Comparing the two lists to see whether there are two basic events in common. If there are two basic events in common, then the task is a contradiction task. Otherwise, the task is not a contradiction task.

   However, note that if the task under consideration contains a module that is logically equivalent to the module, then the task is not a contradiction task. The procedure needs to be carried out on the original fault trees, i.e, before any of the modules are extracted, to ensure that no potential modules are missed.

The task failure fault trees and identified task fault tree modules are converted to BDDs using Equation (2.3.3) and are stored in a library together with the identified module information to support the online analysis. Once the mission configuration is specified in terms of its phases, the relevant BDDs, together with the BDDs for all identified modules, are selected from the library and BDDs for phase failures are combined using the BDD model described in Section 2.5.3.2 to predict the mission unreliability and conditional unreliability of the mission phases. In the global variable ordering for constructing the BDDs for mission failure, all module events are ordered after the basic events [3].

The steps of calculating the conditional unreliability of each mission phase and the mission unreliability using BDD models that include modularisation[3][29] are:

1. (Offline) Construct BDDs for each fault tree representing F_i, for each $i = 1, 2, 3, \cdots, n$ using Equation (2.3.3) under a global variable ordering defined before construction;

identify information for each potential mission module, including its name, equivalent gate, the task it belongs to, lower gates and contradiction tasks as described in this section.

2. (Online) Identify mission modules and construct BDDs representing $F_1 + \cdots + F_i$, for each $i = 2, 3, \cdots, n$ using Equation (2.5.17) and Equation (2.3.3) under the same global variable ordering scheme.

3. (Online) Use Equation (2.5.18) to obtain the probability of each mission module and substitute them into the calculation of the root node of the BDD from step 2; use Equation (2.5.3) and Equation (2.5.2) to calculate the unreliability of conditional phase failures and the whole mission.

2.5.3.3.3 Example Consider the task 1 failure fault tree in Figure 2.5.5, beginning with $G1$ and passing through the tree in a depth-first manner. Gates and events are visited and numbered as shown in Table 2.5.1. Event inputs to any gates are considered before

Step Number	1	2	3	4	5	6	7	8
Node	$G1$	A	B	$G4$	C	D	$G4$	$G1$

Table 2.5.1: Order in which the gates and events are visited in the depth-first traversal of the left most fault tree in Figure 2.5.5

the gate inputs as demonstrated in the table. Each gate is visited at least twice: once on the way down the tree and again on the way back up the tree. A gate that has been visited can be visited again, but the depth-first traversal will not repeated.

The second pass through the tree will find the maximum of the last visits and the minimum of the first visits of the descendants of each gate and the first visit, second visit and last visit step number for each event; and these values are shown in Table 2.5.2.

Gates	$G1$	$G4$	Events	A	B	C	D
Visit 1	1	4	Visit 1	2	3	5	6
Visit 2	8	7	Visit 2	2	3	5	6
Last Visit	8	7	Last Visit	3	4	5	6
Min	2	5					
Max	7	6					

Table 2.5.2: Step number information for the gates and events in task 1 failure fault tree represented in Figure 2.5.5

For this fault tree, gates $G1$ and $G2$ are both modules since each descendant under $G1$ or $G2$ has a first visit step number bigger than the first visit step number of $G1$ or

$G2$ and has a last visit step number smaller than the secondary visit step number of $G1$ or $G2$. The top event of a fault tree is always a module, since it is always the first to be visited first and the last to be visited a second time.

Similarly, the modules of each task fault tree are obtained and shown in Table 2.5.3. $G4$ is equivalent to $G6$ as they have same logic and input events. In the task 3 failure

Module	$M1$	$M2$	$M3$	$M4$	$M5$	$M6$
Module Gate	$G4(G6)$	$G7$	$G5$	$G1$	$G2$	$G3$
Task	1 2	3	2	1	2	3
Lower Gates			$G4(G6)$	$G4(G6)$	$G5$	$G7$
Contradiction Task		1		2 3	1 3	1 2

Table 2.5.3: Module information table for task fault trees represented in Figure 2.5.5

fault tree, gate $G7$ is beneath gate $G3$. Gate $G7$ can be replaced by the module event $M2$, and has one contradiction task, task 1, because event B appears in $M2$ and the fault tree representing the failure of task 1. Gate $G3$ can be replaced by the module event $M6$, it has one lower gate $G7$ and two contradiction tasks, tasks 1 and 2, as A appears in both task failure fault trees. The same process is followed for every module and the results are listed in Table 2.5.3.

After the necessary information for all modules, i.e. all information shown in Table 2.5.3 is obtained, a global ordering including all basic events and module events is defined to allow all modules and task fault trees to be converted into BDDs:

$$A < B < C < D < E < F < M1 < M2 < M3 < M4 < M5 < M6\,. \tag{2.5.19}$$

BDDs for all potential modules are constructed on the premise that all BDDs for modules beneath them have been constructed. First, BDDs for modules that have no below module are constructed. In this case, $M1$ and $M2$ are constructed, and then $M3$, which contains the module $M1$. $M4$, $M5$ and $M6$ are then constructed. The BDDs for the modules and task fault trees with all potential modules are shown in Figure 2.5.11.

If a mission has been defined as represented by the BDDs in Figure 2.5.7, where all the tasks are selected; reviewing Table 2.5.3, it can be seen that only $M1$ and $M3$ are mission modules, since all the other modules have contradiction tasks in the phased mission defined in Figure 2.5.7. The BDDs representing the failure of each phase, F_i, $i = 1, 2, 3$ and mission modules appearing in different phases are represented in Section 2.5.3.1.4 by

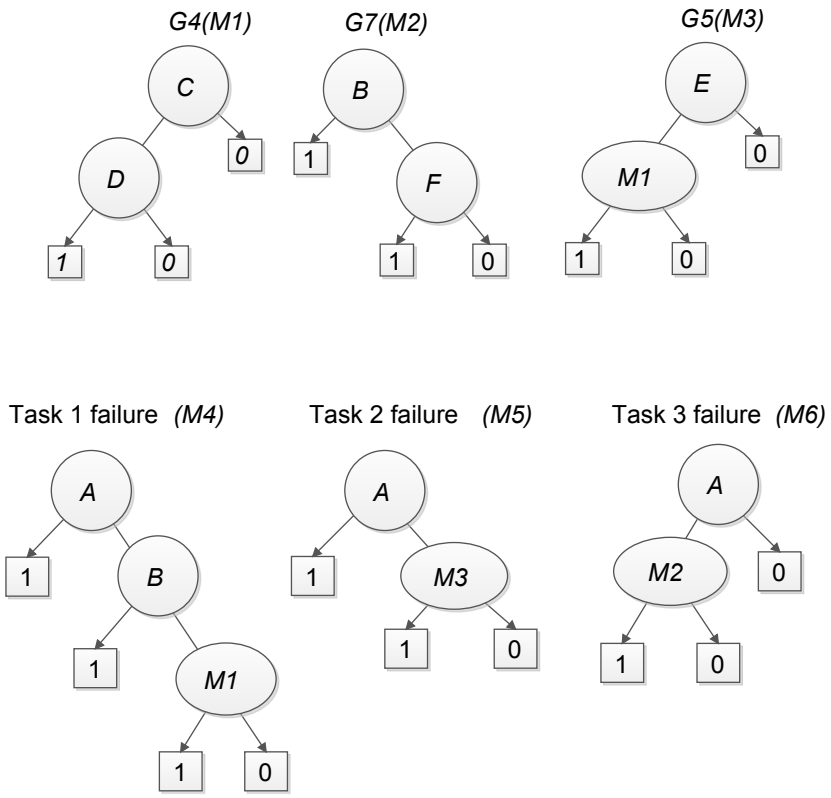

Figure 2.5.11: BDDs for all potential modules in Figure 2.5.5

associating variables in the task BDDs with corresponding phase indices.

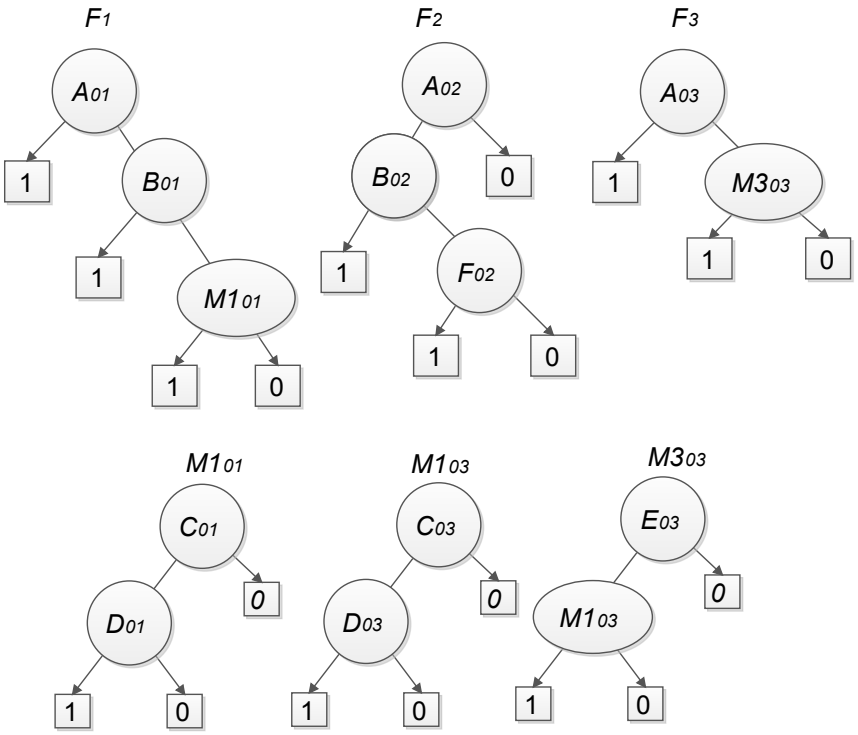

Figure 2.5.12: BDD for F_i

Then, backward PDO in Equation (2.5.17) and the BDD construction rules in Equation (2.3.3) are used to get the BDDs for $F_1 + F_2$ and $F_1 + F_2 + F_3$, which are shown in Figure 2.5.13.

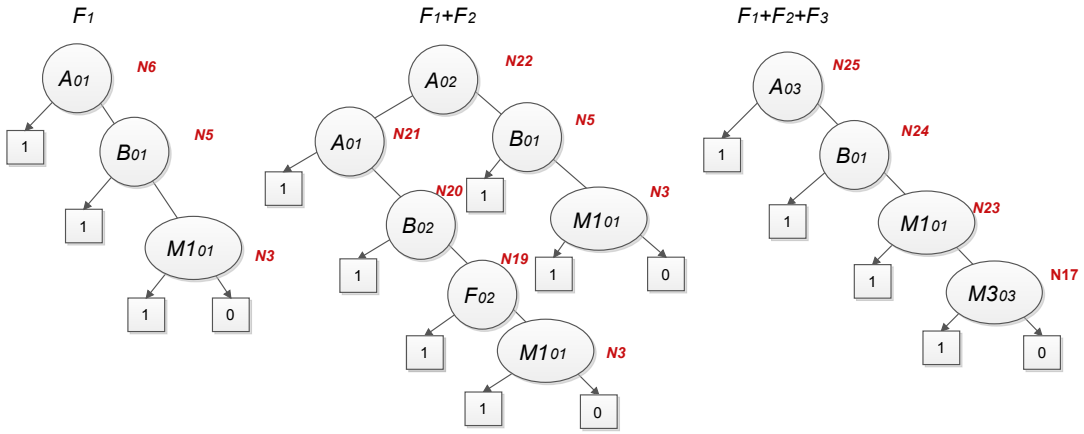

Figure 2.5.13: BDDs for F_1, $F_1 + F_2$ and $F_1 + F_2 + F_3$

Before quantifying the root nodes for F_1, $F_1 + F_2$, and $F_1 + F_2 + F_3$, $N6$, $N22$ and $N25$, the probabilities of the mission module events shown in Figure 2.5.12 need to be

acquired, which are calculated as follows:

$$\begin{aligned} q_{M1_{01}} =& p(C_{01})p(D_{01})\,, \\ q_{M1_{03}} =& p(C_{03})p(D_{03})\,, \\ q_{M3_{03}} =& p(E_{03})p(C_{03})p(D_{03})\,. \end{aligned} \tag{2.5.20}$$

The probabilities of the BDD root node for F_1, F_1+F_2 and $F_1+F_2+F_3$ are calculated using Equation (2.5.18). According to Equation (2.5.2) and Equation (2.5.3), the mission unreliability is given by:

$$Q_{miss} = P(F1 + F2 + F3) = P(N25), \tag{2.5.21}$$

and the conditional phase unreliability are:

$$\begin{aligned} P(ph_1) &= P(F1) = P(N6)\,, \\ P(ph_2) &= P(F1 + F2) - P(F1) = P(N22) - P(N6)\,, \\ P(ph_3) &= P(F1 + F2 + F3) - P(F1 + F2) = P(N25) - P(N22). \end{aligned} \tag{2.5.22}$$

Comparing the BDDs in Figure 2.5.13 and Figure 2.5.10, it can be seen that using module events will simplify the structures of BDDs used in calculating the mission and conditional phase unreliability and thus improve the analysis efficiency. Since this research focuses on the efficiency investigation of reliability analysis models, when choosing the PMS fault trees for testing purposes in the following chapters, fault trees without modules are formed and tested in order to avoid the efficiency impact of the modules and test only the efficiency of the studied analysis techniques.

2.5.4 Reliability Analysis for Multiple Platform PMS

For some phased missions, such as a search and rescue mission, several individual platforms maybe required to work collaboratively in order to achieve an overall mission objective. Such systems are called *multiple platform PMS* and the associated reliability analysis model is described in this section.

2.5.4.1 Introduction to Multiple Platform PMS

A multiple platform phased mission is performed by a number of collaborating individual platforms, each of which must perform its own phased mission with its own objective in

order to achieve the overall mission objective. A typical multiple platform PMS has the following characteristics [4]:

- Individual platforms have the features described in Section 2.4.
- Each platform performs a task or a number of tasks and those tasks performed by a specific platform in a mission are defined as a platform phase. The platform phases contribute to the overall mission objective as part of their own phased mission.
- The platform objectives are not necessarily sequential, i.e, different platform objectives can be carried out in parallel by different platforms, although phases in each single platform must still be performed sequentially.
- Platform objectives can have certain requirements, such as a need for two platform objectives to be carried out in a strict sequence or to start or end simultaneously.
- The mission is assumed to 'fail' if any of the individual platforms fails at any point during the mission.

Individual platforms may be not required to work successfully throughout the whole mission as long as the whole mission objective has been achieved. Individual platforms can start later than the start time of the whole mission and can end earlier than the end time of the whole mission.

2.5.4.1.1 Modelling Failure in Multiple Platform PMS A multiple platform mission definition requires the configuration of each platform phase and the time intervals of mission phases to be known. If the logic expression for the failure of mission phase i, is F_i and the logic expression for the conditional failure of mission phase i is ph_i and m platforms are required to complete the overall mission objective together, F_i is given by the OR combination of failures of all platforms in phase i and is represented by Equation (2.5.23):

$$F_i = F_i^1 + F_i^2 + \cdots + F_i^m, \tag{2.5.23}$$

where F_i^j, $j = 1, 2 \cdots m$ represents the failure in phase i of platform j. For different i or j, F_i^js are independent items since the components of different platforms are independent of each other. When a platform is not active for a certain mission phase, F_i^j is represented by zero.

There are two ways to represent the conditional failure of individual phases and the overall mission failure for multiple platform PMS depending on the different modelling approaches for single platform PMS described in Section 2.5.2.

The first method is achieved by substituting Equation (2.5.23) into Equation (2.5.3) and Equation (2.5.2) in the model described in Section 2.5.2.1 [3]. The critical logic expression that differs from single PMS is given by:

$$\begin{aligned} & F_1 + F_2 + \cdots + F_i \\ =& F_1^1 + F_1^2 + \cdots + F_1^m + F_2^1 + F_2^2 + \cdots + F_2^m + \cdots \\ & + F_3^1 + F_3^2 + \cdots + F_3^m + F_i^1 + F_i^2 + \cdots + F_i^m \,. \end{aligned} \tag{2.5.24}$$

The second method is achieved by substituting Equation (2.5.23) into Equation (2.5.4) and Equation (2.5.5) in the model described in Section 2.5.2.2 [30]. The critical logic expression that differs from single PMS is given by:

$$\begin{aligned} ph_1 &= F_1^1 + F_1^2 + \cdots F_1^m \\ ph_i &= (\overline{F_1^1} \cdot \overline{F_1^2} \cdots \overline{F_1^m}) \cdot (\overline{F_2^1} \cdot \overline{F_2^2} \cdots \overline{F_2^m}) \cdots (\overline{F_{i-1}^1} \cdot \overline{F_{i-1}^2} \cdots \overline{F_{i-1}^m}) \\ & \quad \cdot (F_i^1 + F_i^2 + \cdots F_i^m) \,. \end{aligned} \tag{2.5.25}$$

2.5.4.1.2 Example For example, consider a two platform PMS consisting of a UAV and an autonomous Land Vehicle (LV), with the platform phases and mission phases as shown in Figure 2.5.14 [4]. There are three mission phases for the two platform phased

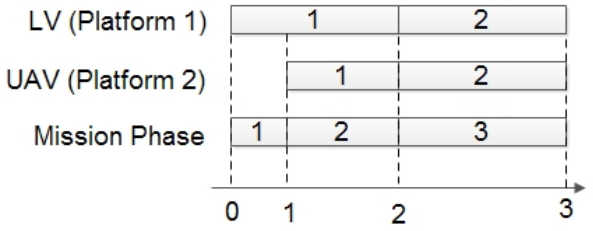

Figure 2.5.14: The individual platform phases and mission phases of the example two platform PMS

mission: Phase 1-the LV performs LV-task 1; Phase 2-the LV continues performing LV-task 1 and the UAV performs UAV-task 1; Phase 3-the LV performs LV-task 2 and the UAV performs UAV-task 2.

Using the first method, the conditional unreliability of phase 1, phase 2 and phase 3 and the mission unreliability are represented in Equation (2.5.26), where F_1^2 is zero, i.e,

at mission phase 1, platform 2 is not active.

$$
\begin{aligned}
P(ph_1) =& P(F_1) = P(F_1^1 + F_1^2) = P(F_1^1)\,, \\
P(F_1 + F_2) &= P(F_1^1 + F_2^1 + F_2^2)\,, \\
P(ph_2) &= P(F_1 + F_2) - P(F_1)\,, \\
Q_{miss} &= P(F_1 + F_2 + F_3) = P(F_1^1 + F_2^1 + F_2^2 + F_3^1 + F_3^2)\,, \\
P(ph_3) &= P(F_1 + F_2 + F_3) - P(F_1 + F_2)\,.
\end{aligned}
\tag{2.5.26}
$$

For the second approach, the unreliability of phase 1 is the same as that found using the first approach. The conditional unreliability of phase 2 and phase 3 and the mission unreliability are represented in Equation (2.5.27).

$$
\begin{aligned}
P(ph_2) &= P(\bar{F_1} \cdot F_2) = P[\overline{F_1^1} \cdot (F_2^1 + F_2^2)]\,, \\
p(ph_3) &= P(\overline{F_1} \cdot \overline{F_2} \cdot F_3) = \overline{F_1^1} \cdot \overline{F_2^1} \cdot \overline{F_2^2} \cdot (F_3^1 + F_3^2)\,, \\
Q_{miss} &= P(ph_1) + P(ph_2) + P(ph_3)\,.
\end{aligned}
\tag{2.5.27}
$$

The probabilities can be calculated using the any of the BDD models described in Section 2.5.3.

2.6 Reliability Analysis as a Decision Making Tool for PMS

2.6.1 Introduction

In order to realise total autonomy, autonomous systems operating phased missions in different environments must be able to interpret their environments and make decisions about their future behaviour without human intervention.

To facilitate a decision making process, two essential tools are used, a diagnostic tool and a prognostic tool [4]. The diagnostic tool is used to identify the changing states of components, subsystems and the corresponding environment and passes this information to the prognostic tool for analysis. The prognostic tool is used to calculate the unreliability of the whole mission objective, as well as the conditional unreliability for individual mission phases, based on all available information provided by the diagnostic tool, for both individual platforms and, in the case of multiple platform PMS, for the overall mission.

Decisions are made based on mission unreliability and there are two times at which these unreliabilities must be calculated as part of a decision making process:

- The initial unreliability is calculated before a mission starts: for a single platform PMS, the prognostic tool needs to perform analysis to measure the conditional unreliability of each mission phase and the mission as a whole.
- The updated unreliability is calculated when a mission is underway and system functionality or environmental changes (for example, engine fan failure or a storm on the planned route of an aircraft flight) are detected by the diagnostic tool: the prognostic tool needs to analyse mission unreliability every time the diagnostic tool provides new information.

For multiple platform PMS, the same initial and updated failure probabilities can be calculated for each single platform performing the mission and also for the overall mission.

In order to use these unreliabilities as part of a decision making tool, an acceptable unreliability is defined for each phase, platform and for the mission as a whole. If the unreliability predicted by the prognostic tool exceeds any of these values then other mission alternatives may need to be considered to make sure the risk of mission failure is not unacceptably high.

The decision making process is shown in Figure 2.6.1.

2.6.1.1 The Calculation of Mission Unreliability

Therefore, for a typical mission, a prognostic assessment would be carried out before the mission starts, which would involve the calculation of the initial unreliability of the original mission configuration. If the unreliability is acceptable, the mission can then be carried out; otherwise, the initial mission configuration needs to be adjusted to an alternative configuration and to be analysed again until the predicted unreliability is acceptable. The initial mission unreliability and conditional phase unreliability is calculated using the methods described in Section 2.5.

When the mission is underway, updated unreliability analysis is carried out by the prognostic tool once information about internal system conditions or environmental changes have been detected by the diagnostic tool; the unreliability is again checked against the acceptable values. If the unreliability is higher than the acceptable values, mission reconfiguration may take place.

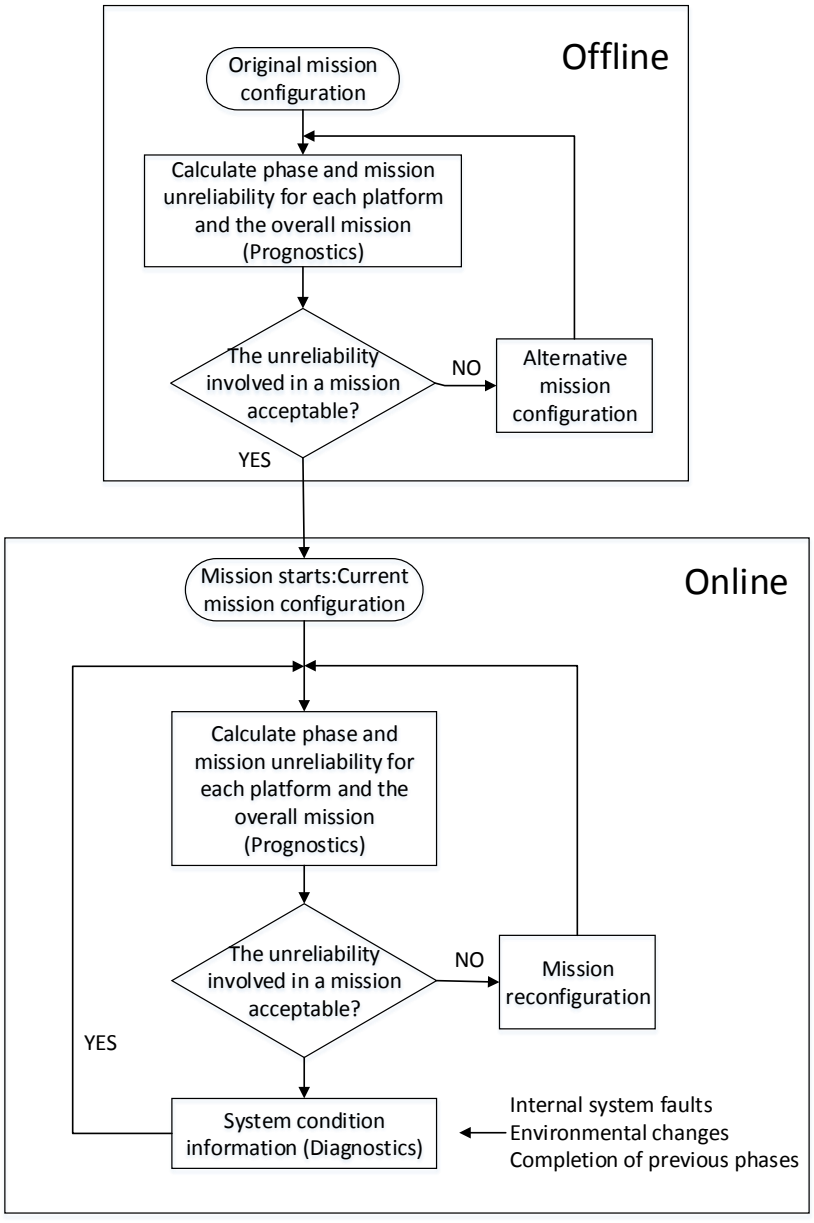

Figure 2.6.1: The decision making process

The calculation of updated mission unreliability when the mission is underway takes the completion of the previous phases into account [1]. The updated conditional unreliability of phase j while considering that previous k phases have been successfully completed, $P(ph_{j|\bar{k}})$, is given by, (using the Bayes' theorem) [4][30]:

$$P(ph_{j|\bar{k}}) = \frac{P(ph_j)}{1 - \sum_{i=1}^{k} P(ph_i)}. \tag{2.6.1}$$

The Q_{miss} is calculated by adding the updated conditional unreliability of the remaining mission phases:

$$Q_{miss|\bar{k}} = \sum_{j=k+1}^{n} P(ph_{j|\bar{k}}). \tag{2.6.2}$$

Using the mission failure modelling technique introduced in Section 2.5.2.1, $P(ph_{j|\bar{k}})$ can be represented as [29]:

$$P(ph_{j|\bar{k}}) = \frac{P(ph_j)}{1 - \sum_{i=1}^{k} P(ph_i)} = \frac{P(F_1 + F_2 + \cdots F_j) - P(F_1 + F_2 + \cdots F_{j-1})}{1 - P(F_1 + F2 + \cdots + F_k)}. \tag{2.6.3}$$

When calculating $P(ph_{j|\bar{k}})$, the $\sum_{i=1}^{k} P(ph_i)$ in Equation (2.6.1) or $P(F_1{+}F2{+}\cdots{+}F_k)$ and $P(F_1 + F_2 + \cdots F_{j-1})$ terms in Equation (2.6.3) are known values obtained from previous analysis, the calculation challenge therefore comes from the determination of the $P(ph_j)$ term in Equation (2.6.1) and $P(F_1 + F_2 + \cdots + F_j)$ in Equation (2.6.3).

2.6.2 Requirements of a Decision Making Process

A key factor in making correct decisions as to the future mission configuration in a changing mission environment is the updated reliability analysis of future mission phases and the mission as a whole. Fast, accurate analysis of future phases and the entire mission could therefore form a crucial element of a decision making process within all systems operating phased missions. It will be particularly important in systems such as UAV, where rapid decisions may be required following system failures. However, the analysis could support decision making in any system. If the unreliability involved in any proposed mission reaches an unacceptable level, then other mission configurations must be considered in order to identify an acceptable alternative. In order to be able to evaluate the available options and select the best possible course of action in a timely manner in a dynamic, rapidly-changing environment where sound decisions must be made quickly,

the fast, accurate reliability analysis of individual phases and the mission is critical. This leads to two main requirements for a reliability-based prognostic tool: accuracy and speed.

It is assumed that a mission is defined well in advance of when it is due to start unless reconfiguration is required during a mission, in which case mission configurations are defined according to the circumstances. Therefore, before a mission starts, there will be enough time to perform the required reliability analysis, meaning that the focus of the reliability analysis is on evaluating accurate (exact) unreliability. When a mission is underway, in order to ensure the system can quickly respond to a changing environment so that the mission can be successfully completed, the updated reliability analysis of the current mission and possible mission alternatives needs to be calculated in the shortest possible time. Accuracy is still important to ensure confidence in any decisions made as a consequence of results obtained using the analysis, meaning that the speed and accuracy of the analysis while the mission is underway is critical. Incorrect or late results may lead to bad decisions being made, which may result in unnecessary mission abortion or serious mission failures.

By converting fault trees that represent the failure logic of a system to BDDs, mission unreliability can be analysed rapidly to give exact values of system unreliability. Therefore, BDDs would appear to be promising for performing the real-time analysis of the system unreliability that is required in a decision making tool. However, work is still required to investigate ways to speed up the analysis. There may be cases when the time available allowed to make a decision is limited and thus it may also be necessary to make a trade off between the accuracy and speed of the reliability analysis. Approximations might be required in order to carry out the reliability analysis of mission alternatives in the shortest possible time. Approximation methods for PMS will then be discussed in Chapter 5.

2.7 Summary

Many systems are required to perform phased missions and to interpret the environment they are in and make decisions about their future behaviour. A key factor in making decisions in changing mission environments is the determination of the reliability of future mission phases and the mission as a whole.

An online-offline strategy, which suggests to carry out as much computation as possible offline before a mission begins and to reduce the amount of online computation required, is proposed in [31] in order to maximise analysis efficiency to support the decision making

process during a mission. This online-offline strategy will be used when investigating the efficiency of models presented in the following chapters.

Many methods have been developed to address dependencies across phases when considering PMS. Fault tree techniques are suitable to model the PMS considered in this research, since the absence of repair processes facilitates the use of such techniques that require an assumption of independence between states of components within the considered systems. Despite the inclusion of multiple failure mode components, which introduce dependencies that make the reliability analysis of PMS significantly more complex than that of systems consisting of a single phase and single failure mode components, the adoption of BDDs enables fast and accurate PMS analysis. Thus the BDD models would appear to offer the greatest chance of performing the real-time reliability analysis that would support a decision making tool for PMS.

There are two ways to model phased missions failures, which are presented in Section 2.5.2. The first modelling technique, which models a system failure occurring between the start of the mission and the end of phase i, $(F_1 + F_2 + \cdots F_i)$ is more efficient during quantitative analysis, since the BDD representing failure up to the previous phase, $F_1 + F_2 + \cdots F_{i-1}$ can be used to form the BDD representing failure up to the current phase, meaning that only the OR operation of $F_1 + F_2 + \cdots F_{i-1}$ and F_i needs to computed. For the second technique, the BDD representing the conditional failure of each mission phase i, $ph_i = \overline{F_1} \cdot \overline{F_2} \cdots F_i$ can not be acquired from the structure of the BDD representing ph_{i-1}, meaning that all operations involved in $\overline{F_1} \cdot \overline{F_2} \cdots F_i$ must be performed when constructing the BDD representing $ph_i = \overline{F_1} \cdot \overline{F_2} \cdots \overline{F_i}$.

Compared with the traditional approach to fault tree analysis, which requires derivation of minimal cut sets for use in the inclusion-exclusion expansion, BDDs are more efficient when analysing PMS. However, because of the dependencies that arise due to the consideration of component failures in different phases and failure modes, the BDD models described in Section 2.3 is no longer suitable and BDD models dealing with these dependencies need to be investigated. Two BDD analysis approaches for PMS are discussed: the first approach constructs BDDs without considering the dependencies which are dealt with during the quantitative analysis. The second approach constructs BDDs while considering the dependencies. Analysis of the two alternatives indicates that over a set of benchmark tests the second approach is the fastest [29] and therefore future investigations in Chapter 3 on BDD models that considers dependencies in PMS containing

multiple failure mode components are based on the second approach.

The BDD models for single platform PMS can be extended to analyse multiple platform PMS by considering the mission phases of each platform individually, as discussed in Section 2.5.4 and the analysis results can then be used to support the decision making process.

The reliability analysis can be used as a decision support tool for PMS containing multiple failure mode components to make decisions on the best next course of actions in a fast-changing environment. In order to make sound decisions in the shortest time possible, the reliability analysis is required to be accurate and fast, which has led to the investigation of BDD-based reliability analysis models and techniques in the following thesis chapters.

Chapter 3

Development of BDD Models for PMS with Multiple Failure Mode Components

3.1 Introduction

The components in a non-repairable PMS exhibit dependencies that make the analysis significantly more complex than that of systems that have a single operational phase. In non-repairable PMS, if a component fails in a certain phase, it will stay in the failed state for all following phases until the end of the mission. Therefore, if a component works in a phase, it must have completed all the previous phases without failing. Components with multiple failure modes also require dependencies to be considered during any system analysis. It is relatively common that one component in a system can fail in more than one failure mode; for example, a valve in an engine might either fail to open, which can lead to no fuel supply, or fail to fully open, which will lead to an insufficient fuel supply, or fail to close, which could lead to an engine fire. All of the failure modes are mutually exclusive, which means, if the valve has failed to open, then it can neither fail to close nor fail partially open for the remainder of the mission.

In order to allow the reliability of PMS to be analysed, it is necessary to be able to account for the above dependencies. Also, to ensure that the reliability analysis methodology could be used to constitute to a real-time decision support tool for systems operating phased missions in changing mission environments, it is necessary that the developed methodology is capable of analysing PMS quickly and accurately. Such a decision support

tool could be used by autonomous systems to help them make timely decisions about the best next course of action when they experience an internal failure or external threat that affects their chance of completing their planned mission, as discussed in Section 2.6.

Although not considering the use of reliability analysis techniques as part of a real-time decision support tool, a number of methods have been developed to address the dependencies that arise during reliability analysis of PMS. Research based on Markov models and Petri nets [40][41][27] considered the cross-phase dependency using state-based approaches. Though the models can deal with dependencies, they face a state space exploration problem as the number of components in a system increases. Methods using fault tree analysis, such as those in [14][42], assume that all components in the system are independent, which simplifies the analysis but is not accurate as dependencies exist due to mission phases and multiple failure modes of components.

Despite the simplifications, fault tree analysis can still not produce results for large systems within a reasonable time frame [2]. This has led to a number of researches to adopt BDDs. [11] presents the advantages of the technique of Boolean function manipulation used by BDD models and [33] proposed the concept of applying the BDDs technique to fault tree analysis. A variety of research has been conducted into the use of BDDs to analyse PMS. The approaches fall into two main categories, as discussed in Section 2.5.3. One approach, presented in [31][30], involves rapidly connecting the BDDs representing individual phase failures together to represent the logic of mission failure without accounting for dependence, which are dealt with during the quantification process. The motivation for this approach was the potential use of the analysis as an input to a decision support tool for autonomous systems undergoing phased missions. The approach allows quantification to begin almost immediately, meaning that information produced during the quantification process is quickly made available to a decision maker. However, the quantification process itself is insufficient due to the fact that dependencies must be accounted for while it is being implemented.

The other approach is to build the BDDs in such a way that these dependencies are considered during construction, meaning that the quantification process is more efficient than in the first approach. Although the analysis developed using approach can be used as inputs to a decision support tool, were not all developed with this application in mind. In [45], researchers developed BDD construction rules using a phase algebra [42] and made changes to the quantification process to analyse the reliability of PMSs for the first

time. Work presented in [3] improves the analysis efficiency of the BDD model in [45] by identifying the modules, subtrees whose basic events do not occur anywhere else in the phased mission fault tree, with module events to simplify the fault tree structure, as discussed in Section 2.5.3. [43][36] extended the BDD model presented in [45] to allow PMSs with multiple failure modes to be analysed. Tests on a set of benchmarks have indicated that the second approach is most efficient [3]. Therefore, further investigation within this thesis will be focus on the second approach.

Phased missions analysis is far more complex than the analysis of missions with single phases because of the involvement of the dependencies between phases and within components. Two BDD models have been developed to take account during construction for the dependencies that arise due to the consideration of component failures in different phases and due to the consideration of multiple failure modes. The DEP-BDD model presented in [43] takes account of these dependencies using an extension of the phase algebra presented in [45]. However, this DEP-BDD model has been shown to be inaccurate in the analysis of PMSs [36][35]. Instead of correcting the DEP-BDD model, the researchers developed a new model, which uses a forward phase and failure mode ordering for BDD construction and an Implicant Tree method to quantify the built BDDs.

In this chapter, the efficiency and accuracy of the existing BDD models for PMS analysis is improved:

1. Two amendment are proposed to the DEP-BDD analysis presented in [43] in order to correct the inaccuracies that have been highlighted in past research.

2. A more efficient quantification method is proposed to replace the Implicant Tree method presented in [36]

The chapter begins by introducing the phase and dependency algebra used to deal with dependencies that arise for components due to their phases of operation and multiple failure modes. The DEP-BDD model [43] is then reviewed and two amendments proposed to correct the observed inaccuracies. This is followed by a review of the BDD model presented in [36] for which a novel quantification method is proposed to replace the Implicant Tree method to improve the analysis efficiency. The performance of the newly-developed BDD models is compared with the previous models by performing tests in which a number of PMS. The chapter ends by summarising these tests and drawing conclusions as to which of these models shows the greatest promise for use within a real-time decision support tool for PMS.

In order to construct BDDs, a variable ordering scheme must initially be defined. Variable ordering schemes for PMS are ordered in these levels: the component level, the phase level and the failure mode level, which will be investigated in Chapter 4. In this chapter, components are ordered as they appear in the fault trees; the phase and failure mode level ordering is performed in order to be in consistent with the applied BDD construction rules.

3.1.1 The Algebra for PMS Analysis With Multiple Failure Modes

In [45], phase algebra is introduced to deal with component dependence across phases and in [43], the researchers extend it to also deal with the dependencies that arise when components have multiple failure modes. A^p_{0i} means components A fails in failure mode p between the start of the mission and the end of phase i, $\overline{A^p_{0i}}$ is the complement of A^p_{0i}, meaning component A does not fail in failure mode p between the start of the mission and the end of phase i. The phase algebra is given in Equation (3.1.1): (suppose $i < j$)

$$
\begin{aligned}
&A^p_{0i} \cdot A^p_{0j} = A^p_{0i} \\
&\overline{A^p_{0i}} \cdot \overline{A^p_{0j}} = \overline{A^p_{0j}} \\
&\overline{A^p_{0i}} \cdot A^p_{0j} = A^p_{ij} \\
&\overline{A^p_{0i}} + \overline{A^p_{0j}} = \overline{A^p_{0i}} \\
&A^p_{0i} + A^p_{0j} = A^p_{0j} \\
&\overline{A^p_{0i}} + A^p_{0j} = 1 \\
&A^p_{0i} \cdot \overline{A^p_{0j}} = 0
\end{aligned}
\tag{3.1.1}
$$

For two variables which relate to the same component but different failure modes, the dependency algebra is given by Equation (3.1.2): (suppose $p \neq q$)

$$
\begin{aligned}
&A^p_{0i} \cdot A^q_{0j} = 0 \\
&\overline{A^p_{0i}} \cdot A^q_{0j} = A^q_{0j} \\
&A^p_{0i} \cdot \overline{A^q_{0j}} = A^p_{0i} \\
&A^p_{0i} + \overline{A^q_{0j}} = \overline{A^q_{0j}} \\
&\overline{A^p_{0i}} + A^q_{0j} = \overline{A^p_{0i}} \\
&A^p_{0i} + A^q_{0j} = A^p_{0i} + A^q_{0j}
\end{aligned}
\tag{3.1.2}
$$

The probabilities of variables relating to the same component after simplification using Equation (3.1.2) are calculated according to the component's failure probability density function if a single term is obtained and Equation (3.1.3) if more than one term is obtained.

$$\begin{aligned} P(\overline{A_{0i}^{p}} \cdot \overline{A_{0j}^{q}}) &= 1 - p(A_{0i}^{p}) - p(A_{0j}^{q}) \\ P(A_{0i}^{p} + A_{0j}^{q}) &= p(A_{0i}^{p}) + p(A_{0j}^{q}) \end{aligned} \tag{3.1.3}$$

Equation (3.1.3) can be extended to more general cases (Equation (3.1.4)): suppose failure mode $p_1 \neq p_2 \neq \cdots p_j$:

$$\begin{aligned} P(\overline{A_{0i_1}^{p_1}} \cdot \overline{A_{0i_2}^{p_2}} \cdots \overline{A_{0i_m}^{p_m}}) &= 1 - p(A_{0i_1}^{p_1}) - p(A_{0i_2}^{p_2}) - \cdots - p(A_{0i_m}^{p_m}) \\ P(A_{0i_1}^{p_1} + A_{0i_2}^{p_2} + \cdots A_{0i_m}^{p_m}) &= \sum_{k=1}^{m} p(A_{0i_k}^{p_k}) \end{aligned} \tag{3.1.4}$$

3.1.2 BDD Model for Systems Containing Components with Multiple Failure Modes

BDD construction requires modification of the standard nodes presented in Section 2.3 if systems contain components with multiple failure modes. Consider for instance the fault tree shown in Figure 3.1.1, which contains basic events relating to three components, A, B and C. Components, A and B, have only one failure mode each, while component C can fail in two different ways, C^1 and C^2. The dependencies between basic events relating to component C must be considered when the BDD is constructed.

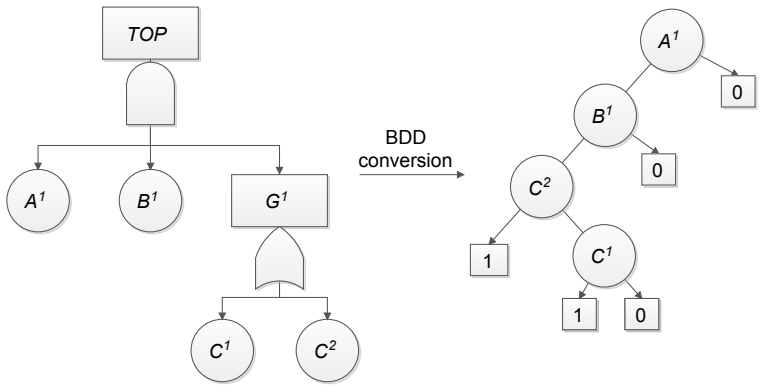

Figure 3.1.1: An example of converting a system containing components with multiple failure modes into BDD

3.1.2.1 BDD Construction Rules

The involvement of components with multiple failure modes requires modification of the BDD rules developed in [33] to deal with the dependencies that arise due to failure modes. The dependencies arise due to the fact that if one component, for example C, fails in failure mode 1, then it cannot fail in any other modes. This is due to the fact the once a component in a non-repairable system fails, it remains in that failed state, meaning that variable relating to the same component but different failure modes are mutually exclusive. The rules of for combining two BDD nodes, F and G, are described in Equation (3.1.5), where $F = ite < x, F1, F0 >$, $G = ite < y, G1, G0 >$ and $x \leq y$ in the variable ordering, and $cp(x)$, $fm(x)$ and $pn(x)$ are the component to which x relates, its failure mode and the indices of the phases within which x occurs, respectively.

$$F \Diamond G = \begin{cases} ite < x, F1 \Diamond G1, F0 \Diamond G0 > & \text{when } x = y \\ ite < x, F1 \Diamond G0^*, F0 \Diamond G > & \text{when } cp(x) = cp(y) \text{ and } fm(x) \neq fm(y) \\ ite < x, F1 \Diamond G, F0 \Diamond G > & \text{when } cp(x) \neq cp(y). \end{cases} \quad (3.1.5)$$

- $G0^* = (G)_{x=1} = (0 \cdot G1 + 1 \cdot G0)_{x=1} = (G0)_{x=1}$ is the first node with a variable relating to a different component than x encountered during the traversal down the 0-branch of node G. This is because when x and y, relates to different failure modes of the same component, traversing the 1-branch from node F means that $fm(x)$ has occurred, meaning that $fm(y)$ cannot have occurred, thus meaning that G must be traversed on its 0-branch only.

3.1.2.2 Example

Consider the example fault tree shown in Figure 3.1.1, and suppose that $A < B < C$ and failure modes are ordered in a backward fashion. The variable ordering is therefore: $A^1 < B^1 < C^2 < C^1$ and the BDD is constructed as follows. For gate $G1$,

$$ite < C^2, 1, 0 > + ite < C^1, 1, 0 > = ite < C^2, 1, ite < C^1, 1, 0 >>$$

For the TOP gate, events A^1 and B^1 are first considered:

$$ite < A^1, 1, 0 > \cdot ite < B^1, 1, 0 > = ite < A^1, ite < B^1, 1, 0 >, 0 >$$

and then this BDD structure is connected to the one obtained for gate $G1$:

$$\begin{aligned} TOP =& ite < A^1, ite < B^1, 1, 0 >, 0 > \cdot ite < C^2, 1, ite < C^1, 1, 0 >> \\ =& ite < A^1, (ite < B^1, 1, 0 > \cdot ite < C^2, 1, ite < C^1, 1, 0 >>), 0 > \\ =& ite < A^1, (ite < B^1, ite < C^2, 1, ite < C^1, 1, 0 >>, 0 >, 0 >), 0 > \end{aligned}$$

The BDD constructed is shown on the right in Figure 3.1.1.

3.2 The DEP-BDD Model and Its Improvement

This section reviews the DEP-BDD model developed in [43], which was shown to be inaccurate in the analysis of PMS[35][36]. Two amendments are proposed to the DEP-BDD analysis that correcting the previously observed inaccuracies.

3.2.1 BDD Construction

The DEP-BDD model uses backward phase ordering and backward failure mode ordering and considers the phase level before the failure mode level. To illustrate this, consider a system capable of performing three tasks, the failure of which is represented by the fault tree shown in Figure 3.2.1.

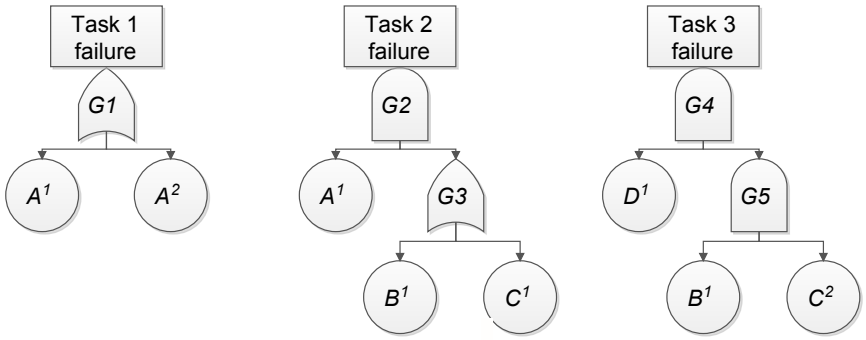

Figure 3.2.1: Fault trees representing the failures of three tasks a system capable of

Suppose the system is required to perform a mission in which task 1 and task 2 are performed in sequence. In order to apply the DEP-BDD model, the variables would be ordered as follows: $A^1_{02} < A^2_{01} < A^1_{01} < B^1_{02} < C^1_{02}$.

The DEP-BDD model computes the operation between two nodes $F = ite < x, F1, F0 >$

and $G = ite < y, G1, G0 >$ using Equation (3.2.1). Suppose $x \leq y$,

$$F \diamond G = \begin{cases} ite < x, F1 \diamond G1, F0 \diamond G0 > & \text{when } x = y \\ ite < x, F1 \diamond G, F0 \diamond L0 > & \text{when } cp(x) = cp(y) \text{ and } fm(x) = fm(y) \\ ite < x, F1 \diamond L1, F0 \diamond G > & \text{when } cp(x) = cp(y) \text{ and } fm(x) \neq fm(y) \\ ite < x, F1 \diamond G, F0 \diamond G > & \text{when } x \neq y. \end{cases} \tag{3.2.1}$$

where

- $L1 = (G0)_{x=1}$, $L0 = (G0)_{x=0}$ is the first node with variable relating to a component other than x encountered during a traversal down the 0-branches of the BDD starting from G.

3.2.2 Quantitative Analysis

To demonstrate how to quantify the probability of a node G, suppose $G = ite < x, G1, G0 >$, $G1 = ite < y, H1, H0 >$ and $G0 = ite < z, I1, I0 >$, as shown in Figure 3.2.2.

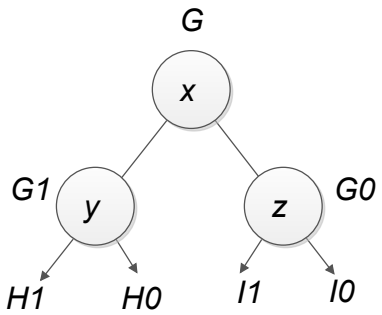

Figure 3.2.2: General structure of a node constructed using the DEP-BDD rules

For a BDD node in a DEP-BDD model, the 1-branch always links two variables that relate to the same component and failure mode but with different phase index, or that relate to different components; the 0-branch always links two variables that relate to different failure modes of the same component, or that relate to different components.

According to the relations between $cp(x)$ and $cp(y)$, and $cp(z)$; $fm(x)$, $fm(y)$, and $fm(z)$; $pn(x)$, $pn(y)$, and $pn(z)$ of the three variables x, y and z, the probability of BDD

node G is calculated by Equation (3.2.2).

$$P(G) = \begin{cases} p(x)P(G1) + [1 - p(x)]P(G0) & \text{case 1} \\ P(G1) + P(G0) - P(H0) + p(x)[P(H0) - P(G0)] & \text{case 2} \\ P(G0) + p(x)[P(G1) - P(I0^*)] & \text{case 3} \\ P(G1) + P(G0) - P(H0) + p(x)[P(H0) - P(I0^*)]. & \text{case 4} \end{cases} \tag{3.2.2}$$

where $I0^* = (I0)_{x=1}$ are the first node with variable relating to a different component to x encountered during the traversal down the 0-branch of node $G1$ and $G0$ respectively. The cases for the relationships between x, y and z are:

case 1 $cp(x) \neq cp(y)$ and $cp(x) \neq cp(z)$.

case 2 $cp(x) = cp(y)$, $fm(x) = fm(y)$ and $cp(x) \neq cp(z)$.

case 3 $cp(x) \neq cp(y)$ and $cp(x) = cp(z)$, $fm(x) \neq fm(z)$.

case 4 $cp(x) = cp(y)$ and $cp(x) = cp(z)$, $fm(x) = fm(y)$, $pn(x) \neq pn(y)$ and $fm(x) \neq fm(z)$.

3.2.3 Analytical Inaccuracies

In [35] and [36], it was pointed out that the DEP-BDD analysis presented above is inaccurate. This inaccuracies can be demonstrated by means of a simple example. A PMS is considered where two tasks are performed in sequence and have failure logic represented by the fault trees relating to task 1 and task 2 in Figure 3.2.1. Analysis is first performed using a conventional approach as discussed in Section 2.5.2.1. Next, the DEP-BDD method given in Equation (3.2.2) is applied and the inaccuracies in the approach is noted.

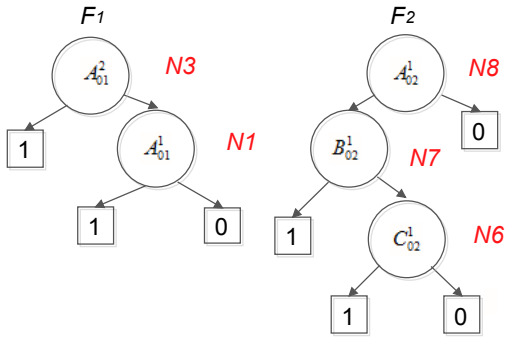

Figure 3.2.3: BDD for each phase failure F_i

3.2.3.1 Inaccuracies in Construction

Using the technique described in [44], the exact conditional unreliability for each phase and the overall mission unreliability are calculated as follows:

$$\begin{aligned} P(ph_1) &= P(F_1) = P(A^1_{01} + A^2_{01}) = p(A^1_{01}) + p(A^2_{01}) \\ P(ph_2) &= P(A^1_{02} \cdot B^1_{02} + A^1_{02} \cdot C^1_{02}) - P[(A^1_{02} \cdot B^1_{02} + A^1_{02} \cdot C^1_{02}) \cdot (A^1_{01} + A^2_{01})] \\ &= [p(A^1_{02}) - p(A^1_{01})] \cdot [p(B^1_{02}) + p(C^1_{02}) - p(B^1_{02}) \cdot p(C^1_{02})] \\ Q_{miss} &= P(ph_1) + P(ph_2) \\ &= p(A^1_{01}) + p(A^2_{01}) \\ &+ [p(A^1_{02}) - p(A^1_{01})] \cdot [p(B^1_{02}) + p(C^1_{02}) - p(B^1_{02}) \cdot p(C^1_{02})]. \end{aligned} \tag{3.2.3}$$

Using the DEP-BDD construction rules, BDD representing the condition for mission failure is built by combining the BDDs for failure in each phase, shown in Figure 3.2.3, to model an expression for $F_1 + F_2$. The construction is presented in Equation (3.2.4) and the relevant BDD is shown in Figure 3.2.4.

$$\begin{aligned} & F_1 + F_2 = N3 + N8 \\ =& ite < A^1_{02}, N7 + 0, 0 + N3 > \\ =& ite < A^1_{02}, N7, N3 > \\ =& ite < A^1_{02}, ite < B^1_{02}, 1, ite < C^1_{02}, 1, 0 >>, ite < A^2_{01}, 1, ite < A^1_{01}, 1, 0 >>> . \end{aligned} \tag{3.2.4}$$

The root node of the BDD representing the condition for mission failure is $N17$. Traversing the BDD form $N17$ along its 0-branch, it can be seen that the variable of node $N1$, A^1_{01}, relates to the same failure mode of the same component as the variable

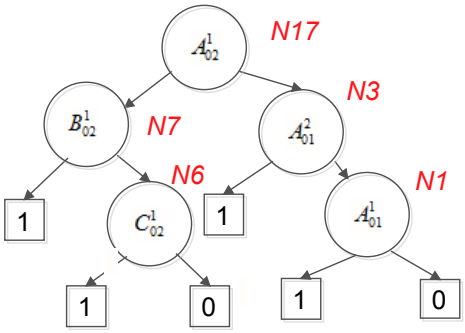

Figure 3.2.4: BDDs For $F_1 + F_2$ using DEP-BDD rules

of the root node $N17$, A^1_{02}. This means that $N1$ is a redundant node because $A^1_{02} = 0$ implies $A^1_{01} = 0$, since if A does not fail in failure mode 1 between the start of the mission and the end of phase 2, it necessarily does not fail in failure mode 1 in phase 1. The redundant node unnecessarily increase the BDD size and hence the time taken to perform the analysis.

3.2.3.2 Inaccuracies in Quantification

The possible inclusion of redundant nodes within the BDD can also lead to inaccuracies in the quantitative analysis of the BDD [36][35]. This is demonstrated by applying the quantification rules presented in Equation (3.2.2) to calculate the probability of node $N17$:

$$\begin{aligned} P(N17) =& P(G0) + p(x) \cdot [P(G1) - P(I0^*)] \quad \text{(case 3)} \\ =& P(N3) + p(A^1_{02}) \cdot [P(N7) - 0] \\ =& P(N1) + p(A^2_{01})[1 - 0] \quad \text{(case 3)} \\ &+ P(A^1_{02}) \cdot [[p(B^1_{02}) + p(C^1_{02}) - p(B^1_{02}) \cdot p(C^1_{02})]] \quad \text{(case 1)} \\ =& p(A^1_{01}) + p(A^2_{01}) + p(A^1_{02}) \cdot [p(B^1_{02}) + p(C^1_{02}) - p(B^1_{02}) \cdot p(C^1_{02})]. \end{aligned} \tag{3.2.5}$$

It can be seen that this result is different from the mission unreliability calculated in Equation (3.2.3). This is partly due to the fact that node $N1$ is redundant but a part is also played by the method of identifying $L1$ (the first node with variable relating to a component other than A during a traversal down the 0-branches starting from $N17$). The BDD construction (using Equation (3.2.1)) does not account for variables relating to different failure modes of the same component as the root node during the traversal down 0-branches from the root node.

3.2.4 Improving the DEP-BDD Model

The DEP-BDD rules are shown to lead to inaccurate quantification due to the incorrect formulation of $L1$ and $L0$ and inclusion of redundant nodes [35]. However, instead of attempting to correct the DEP-BDD rules, new phase BDD rules are developed in [35][36].

In this section, two modifications are proposed in order to correct the analysis by eliminating redundant nodes in the BDD and hence allowing accurate quantification to be performed. The first modification involves adding a reduction process during construction and the second, alternative, modification involves amending the variable ordering.

The DEP-BDD quantification process described in Section 3.2.2 is used to quantify the two modified models presented in this section, which are referred to as Model 1 and Model 2.

3.2.4.1 DEP-BDD Analysis with a Reduction Process (Model 1)

Model 1 retains the variable ordering of the DEP-BDD model and corrects the quantification process by amending the formulation of $L1$ and $L0$. A reduction process is also added to simplify the construction of two nodes when variables relate to the same component and failure mode. The amendments related to $L1$ and $L0$ ensure the correct nodes are found during traversal down the 0-branch of G while the reduction process ensures BDDs remain in a compact format and are ready for DEP-BDD quantification.

The formulation of $L1$ and $L0$ in Equation (3.2.1) are modified to:

- $L1 = (G0)_{x=1}$ is the first node with a variable relating to a component other than x or relating to the same component and failure mode as x encountered on the traversal down the 0-branches of the node starting from G. This is because $x = 1$ implies variables related to another failure mode of the same component must be equal to 0 whereas this is not the case for other cases.

- $L0 = (G0)_{x=0} = G0$. In DEP-BDD analysis, the 0-branch of node G always links two variables that relate to different failure modes of the same component or that relate to different components. When $x = 0$, $L0 = G0$ always applies.

The **reduction process** is carried out when computing the combination of two nodes whose variables relate to identical components but different failure modes (or operation $(F_1+F_2+\cdots F_m)+F_{m+1}$ is performed). The process involves traversing down the 0-branch

of the newly-created node and replacing any node with a variable relating to an identical component and failure mode as the newly-created node by its 0-branch.

The theoretical proof of the the reduction process is given below. The redundant node could appear on both 1-branch and 0-branch of the new created node. We can simply consider the redundant node on the 0-branch, since the sub-node sharing property of BDDs will allow all the same redundant nodes to be reduced.

Proof 3.2.1 *Suppose* $G = ite < x, G1, G0 >$, $G1 = ite < y, H1, H0 >$, *and* $G0 = ite < z, I1, I0 >$, *as shown in Figure 3.2.5 . Since the reduction process operates on nodes on*

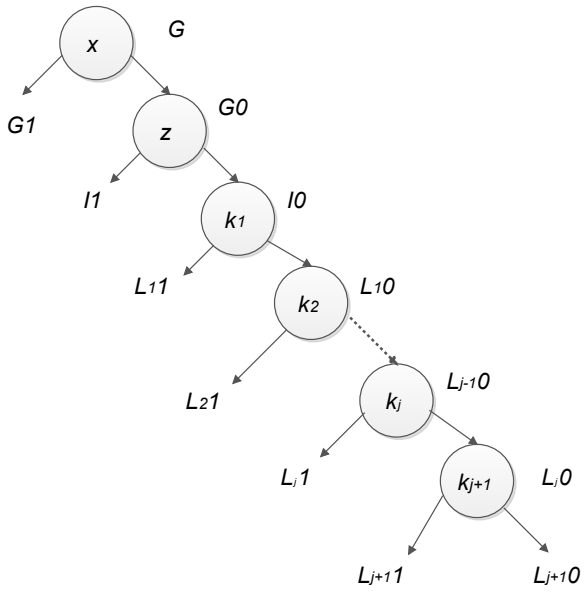

Figure 3.2.5: The BDD node illustration for reduction proof

the success branch of G *only, assume without loss of generality that the variable of* $G1$, y, *meets the condition that* $cp(x) \neq cp(z)$. *Then,*

$$P(x \cdot G1 + \bar{x} \cdot G0) = p(x) \cdot P(G1) + P(\bar{x} \cdot G0).$$

Suppose on the success branch of G0, there is a node $L_j0 = ite < k_{j+1}, L_{j+1}1, L_{j+1}0 >$, *whose variable* k_{j+1} *satisfies the condition* $cp(x) = cp(k_{j+1})$, $fm(x) = fm(k_{j+1})$ *and* $pn(x) > pn(k_{j+1})$.

$$\begin{aligned} & P(\overline{x} \cdot G0) \\ =& P(\overline{x} \cdot (z \cdot I1 + \overline{z} \cdot I0)) \end{aligned}$$

$$
\begin{aligned}
&= P(\overline{x} \cdot (z \cdot I1 + \overline{z} \cdot (k_1 \cdot L_1 1 + \overline{k_1} \cdot L_1 0))) \\
&= P(\overline{x} \cdot (z \cdot I1 + \overline{z} \cdot (k_1 \cdot L_1 1 + \overline{k_1} \cdot (k_2 \cdot (\cdots L_{j-1} 1) \\
&\quad + \overline{k_2} \cdot (\cdots k_j \cdot L_j 1 + \overline{k_j} \cdot (k_{j+1} \cdot L_{j+1} + \overline{k_{j+1}} \cdot L_{j+1} 0)))))))
\end{aligned}
$$

Since x and k_j relate to the same component and failure mode and $pn(x) > px(k_{j+1})$, $\overline{x} \cdot k_{j+1} = 0$ (according to the last equation of Equation (3.1.1)),

$$
\begin{aligned}
&= P(\overline{x} \cdot z \cdot I1 + \overline{x} \cdot \overline{z} \cdot (k_1 \cdot L_1 1 + \overline{k_1} \cdot (k_2 \cdot (\cdots L_{j-1} 1) + \\
&\quad \overline{x} \cdot \overline{k_2} \cdot (\cdots + k_j \cdot L_j 1 + \overline{k_j} \cdot \overline{k_{j+1}} \cdot L_{j+1} 0)))),
\end{aligned}
$$

Since x and k_{j+1} relate to the same component and failure mode and $pn(x) > px(k_{j+1})$, $\overline{x} \cdot \overline{k_{j+1}} = \overline{x}$ (according to the second equation of Equation (3.1.1)),

$$
\begin{aligned}
&= P(\overline{x} \cdot z \cdot I1 + \overline{x} \cdot \overline{z} \cdot (k_1 \cdot L_1 1 + \overline{k_1} \cdot (k_2 \cdot (\cdots L_{j-1} 1) + \\
&\quad \overline{x} \cdot \overline{k_2} \cdot (\cdots + k_j \cdot L_j 1 + \overline{k_j} \cdot L_{j+1} 0))))) \\
&= P(\overline{x} \cdot (z \cdot I1 + \overline{z} \cdot (k_1 \cdot L_1 1 + \overline{k_1} \cdot (k_2 \cdot (\cdots L_{j-1} 1) + \\
&\quad \overline{k_2} \cdot (\cdots + k_j \cdot L_j 1 + \overline{k_j} \cdot L_{j+1} 0)))))
\end{aligned}
$$

From this expression, it can be seen that node $L_j 0$ has been replaced by its 0-branch $L_{j+1} 0$.

3.2.4.1.1 Example Consider the mission defined as in Figure 3.2.3. Using model 1 , the BDD for $F_1 + F_2$ is computed in Equation (3.2.6):

$$
\begin{aligned}
&F_1 + F_2 = N3 + N8 = ite < x, F1 \diamond L1, F0 \diamond G > \\
&= ite < A^1_{02}, N7 + N1, 0 + N3 > \\
&= ite < A^1_{02}, ite < A^1_{01}, 1, N7 >, N3 > \\
&= ite < A^1_{02}, ite < A^1_{01}, 1, ite < B^1_{02}, 1, ite < C^1_{02}, 1, 0 >>>, ite < A^2_{01}, 1, ite < A^1_{01}, 1, 0 >>>
\end{aligned}
\tag{3.2.6}
$$

where $L1 = N1$, as A^1_{01} and A^1_{02} relate to the same component and failure mode.

The BDD for $F_1 + F_2$ before reduction is shown on the left of Figure 3.2.6. On the traversal down the 0-branch of the newly-created node, $N17$, the variable of $N1$ shares the same component and failure mode with the variable of $N17$, therefore, node $N1$ is replaced by its 0-branch, 0. The BDD for $F_1 + F_2$ after reduction is therefore as shown

on the right of Figure 3.2.6.

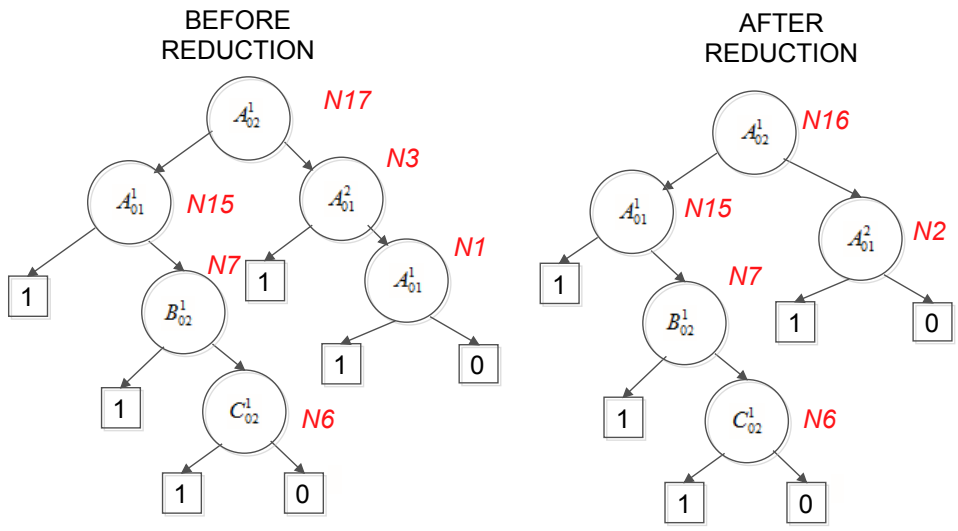

Figure 3.2.6: BDDs For $F_1 + F_2$ before and after reduction

The reduction process eliminates the redundant node, $N1$, in the BDD, keeping the BDD in a simplified form. This allows for accurate quantification and also pointing benefits the analysis speed due to the reduction in the number of nodes in the BDD.

The necessary of the reduction process is emphasised in the following quantification process.

Applying the DEP-BDD quantification rules given in Equation (3.2.2), the probability of node $N16$ is calculated as follows:

$$\begin{aligned}
P(N16) =& P(G1) + P(G0) - P(H0) + p(x) \cdot [P(H0^*) - p(I0^*)] \quad \text{(case 4)} \\
=& P(N15) + P(N2) - P(N7) + p(A^1_{02}) \cdot [P(N7) - 0] \\
=& p(A^1_{01}) + [1 - p(A^1_{01})] \cdot P(N7) + P(N2) \\
& - P(N7) + p(A^1_{02}) P(N7) \quad \text{(case 1)} \\
=& p(A^1_{01}) + p(A^2_{01}) + [p(A^1_{02}) - p(A^1_{01})] \cdot P(N7) \\
=& p(A^1_{01}) + p(A^2_{01}) \\
& + [p(A^1_{02}) - p(A^1_{01})] \cdot [p(B^1_{02}) + p(C^1_{02}) - p(B^1_{02}) \cdot p(C^1_{02})]
\end{aligned} \tag{3.2.7}$$

The calculated mission unreliability is identical to that given in Equation (3.2.3), and demonstrates how the amended $L1$ and $L0$ searching method and the reduction process have corrected the errors in the DEP-BDD analysis model presented in [43].

3.2.4.2 DEP-BDD Analysis with Amending Variable Ordering (Model 2)

Model 2 corrects the DEP-BDD analysis by changing the variables ordering and adopting the same $L1$ and $L0$ calculation as Model 1. Model 2 requires the variables to be ordered firstly according to failure mode level and then phase level. This is the only change from the Model 1 analysis. However, by considering failure mode level first, the phase dependency operation automatically eliminate redundant nodes from the BDD. No additional reduction process is needed.

3.2.4.2.1 Example Consider again the mission consisting of two phases, where conditions for failure are modelled by the BDDs shown in Figure 3.2.3. The variable ordering list required by Model 2 is: $A^2_{01} < A^1_{02} < A^1_{01} < B^1_{02} < C^1_{02}$. A^2_{01} is listed earlier than A^1_{02} compared with DEP-BDD analysis, because Model 2 considers dependencies between failure modes before dependencies between phases. By applying the DEP-BDD rules and using the $L1$ and $L0$ searching method described for Model 1, the BDD for $F_1 + F_2$ is constructed using Model 2 as shown in Equation (3.2.8).

$$\begin{aligned} F_1 + F_2 &= N3 + N8 = ite < x, F_1 \diamond L1, F0 \diamond G > \\ &= ite < A^2_{01}, 1 + 0, N1 + N8 > \\ &= ite < A^2_{01}, 1, ite < A^1_{02}, N1 + N7, 0 + 0 >> \\ &= ite < A^2_{01}, 1, ite < A^1_{02}, ite < A^1_{01}, 1, ite < B^1_{02}, 1, ite < C^1_{02}, 1, 0 >>>, 0 >> \end{aligned} \tag{3.2.8}$$

where $L1 = 0$ in this case. The BDD for $F_1 + F_2$ obtained using Model 2 is shown in Figure 3.2.7

The figure illustrates that there is no variable in the success branch of node $N17$ that shares the same component and failure mode with the variable of $N17$, due to the fact that the BDDs constructed under this variable ordering automatically eliminate the redundant nodes.

Using case3 Equation (3.2.2), the BDD constructed using Model 2 is quantitatively

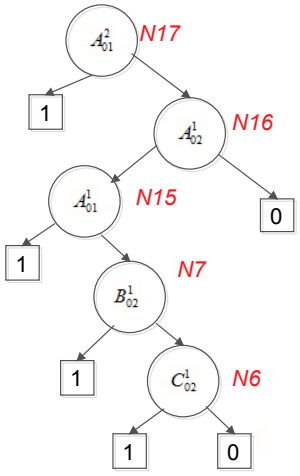

Figure 3.2.7: BDD For $F_1 + F_2$ constructed using Model 2

analysed in Equation (3.2.9):

$$\begin{aligned} P(N17) =& P(G0) + p(x) \cdot [P(G1) - p(I0^*)] \quad \text{case 3} \\ =& P(N16) + p(A^2_{01}) \cdot [1 - 0] \\ =& P(N15) + 0 - P(N7) + p(A^1_{02}) \cdot P(N7) + p(A^2_{01}) \quad \text{case 2} \\ =& p(A^1_{01}) + p(A^2_{01}) + [p(A^1_{02}) - p(A^1_{01})] \cdot P(N7) \\ =& p(A^1_{01}) + p(A^2_{01}) + [p(A^1_{02}) - p(A^1_{01})] \cdot [p(B^1_{02}) + p(C^1_{02}) - p(B^1_{02}) \cdot p(C^1_{02})] \end{aligned} \tag{3.2.9}$$

This is in agreement with the results shown in Equation (3.2.3) and Equation (3.2.7), demonstrating the fact that Model 2 can be used to correctly perform an analysis.

3.3 The Forward-BDD Model

Rather than attempting to correct the errors in the analysis performed in the DEP-BDD model [43], Researchers in [36] developed an alternative BDD analysis technique that uses an Implicant Tree-based method for quantification. The model is denoted as Forward-BDD because it orders variables using a forward phase index. It also considers failure mode level before phase level when ordering variables.

3.3.1 BDDs Construction Rules

The Forward-BDD model computes the operation between two BDD nodes, $F = ite < x, F_1, F0 >$ and $G = ite < y, G1, G0 >$, using Equation (3.3.1). Suppose that $x \leq y$:

$$F \diamond G = \begin{cases} ite < x, F1 \diamond G1, F0 \diamond G0 > & x = y \\ ite < x, F1 \diamond G0^*, F0 \diamond G > & cp(x) = cp(y), fm(x) \neq fm(y) \\ ite < x, F1 \diamond G1, F0 \diamond G > & cp(x) = cp(y), fm(x) = fm(y) \\ ite < x, F1 \diamond G, F0 \diamond G > & cp(x) \neq cp(y) \end{cases} \tag{3.3.1}$$

Explanation of this Rule:

1. $G0^* = (G0)_{x=1}$ is the first node with variable relating to a component other than x encountered during a traversal down the 0-branch of the BDD starting from G.

For a newly-created node as shown in Figure 3.2.2, a **reduction rule** is introduced to remove redundant nodes. If $cp(x) = cp(z)$, $fm(x) = fm(z)$, and $G1 = I1$, then node G is replaced by its success branch $G0$. This is because under the variable ordering scheme applied in the Forward-BDD model, $x = 1$ implies $z = 1$ and therefore, $G1$ and $I1$ are the same node.

3.3.1.0.2 Example Consider for instance the mission defined by the BDDs shown in Figure 3.2.3. Variables relating to the same component but different failure modes are ordered in a backward fashion when constructing the BDDs for the task fault trees. Therefore, when applying the Forward-BDD model, variables must be first ordered by backward failure mode and then forward phase index. With component order $A < B < C$, variables relating to the same component but with bigger failure modes are ordered before those with smaller failure modes. For variables with the same component and failure mode, those with smaller phase index are ordered earlier. The variable ordering scheme is therefore: $A^2_{01} < A^1_{01} < A^1_{02} < B^1_{02} < C^1_{02}$.

The BDD representing $F_1 + F_2$ is computed using the Forward-BDD model as in

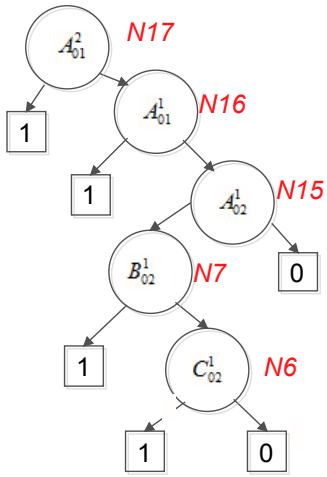

Figure 3.3.1: BDD For $F_1 + F_2$

Equation (3.3.2):

$$\begin{aligned} & F_1 + F_2 = N3 + N8 = ite < x, F_1 \diamond G0^*, F0 \diamond G > \\ =& ite < A^2_{01}, 1 + 0, N1 + N8 > \\ =& ite < A^2_{01}, 1, ite < A^1_{01}, 1 + N7, 0 + N8 >> \\ =& ite < A^2_{01}, 1, ite < A^1_{01}, 1, ite < A^1_{02}, ite < B^1_{02}, 1, ite < C^1_{02}, 1, 0 >>, 0 > \end{aligned} \tag{3.3.2}$$

where $G0^* = 0$. The BDD representing $F_1 + F_2$ is shown in Figure 3.3.1.

3.3.2 The Implicant Tree Method

The Implicant Tree method developed in [36] allows quantification of the BDDs constructed using the Forward-BDD model by constructing a dependency free data structure, the Implicant Tree, which is used for quantification. The Implicant Tree is an acyclic graph with two types of nodes: Event nodes, which have a single child node, and represent the failure and success of basic events; Summing nodes which have multiple child nodes relating to the same component and represent an OR logic combination of these child nodes. The Implicant Tree is formed in accordance with the BDD structure in a bottom-up manner using a set of phase dependence algebra.

This method ensures that each component appears only once on a path from the top node to the terminal nodes in the Implicant Tree and therefore all variables in a path are independent to each other so that the following quantification can be rapidly conducted.

3.3.2.1 The Dependency Rules

During construction of the Implicant Tree, a set of rules derived from the algebra presented in Equation (3.1.1) and Equation (3.1.2) is used to address the dependencies between variables relating to the same component but having different failure modes or phase index. The rules are listed in Equation (3.3.3) and Equation (3.3.4).
For variables relating to same failure mode and same component, supposing $i < j$:

$$\begin{aligned} p(\bar{A^p_{0i}} \cdot A^p_{0j}) &= p(A^p_{ij}) \\ p(\bar{A^p_{0i}} \cdot \bar{A^p_{0j}}) &= p(\bar{A^p_{0j}}) \end{aligned} \tag{3.3.3}$$

For variables relating to the same component but different failure modes, suppose $p \neq q$:

$$\begin{aligned} &p(\bar{A^p_{0i}} \cdot A^q_{jk}) = p(A^q_{jk}) \\ &p(\bar{A^{q_1}_{0i_1}} \cdot \bar{A^{q_2}_{0i_2}} \cdots \bar{A^{q_n}_{0i_n}}) = 1 - p(A^{q_1}_{0i_1}) - p(A^{q_2}_{0i_2}) - \cdots - p(A^{q_n}_{0i_n}) \end{aligned} \tag{3.3.4}$$

The algebra in Equation (3.3.3) and Equation (3.3.4) is used to solve the dependencies across variables relating to the same component so that one component appears only once on a Implicant Tree path and therefore independency between nodes on the path can be achieved.

3.3.2.2 Construction of the Implicant Tree

The Implicant Tree representing the BDD is constructed starting from the terminal node of the BDD. The Implicant Tree contains two types of nodes.

Event node is represented by Equation (3.3.5):

$$IT(G) = IT < x, child >, \tag{3.3.5}$$

where x is the occurrence or non-occurrence of an event (failure or success operation of a component), and *child* is an IT node, which could be terminal 1 or an event node or an summing node. Particularly, the event node for BDD terminal node 1 is represented by: $IT(1) = 1$, whose probability is 1 and the event node for a BDD terminal node 0 is represented by: $IT(0) = NULL$, whose probability is 0; $NULL$ represents the ITnode doesn't exist, meaning that 0 cannot contribute to system failure. An example is illustrated in Figure 3.3.2, where $x = \overline{B^1_{02}}$ standing for the success of event B^1_{02} and the child is any other IT node.

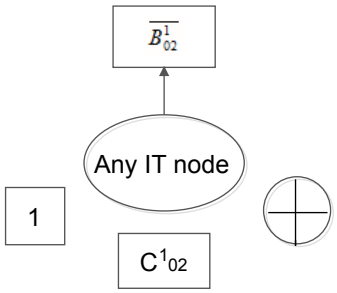

Figure 3.3.2: An event node

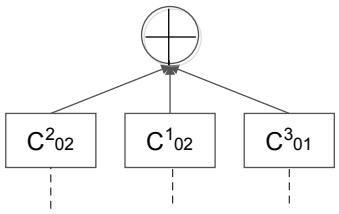

Figure 3.3.3: An sum node

Summing node suppose there are n child branches for this summing node, which is represented by Equation (3.3.6):

$$IT(G) = IT < \bigoplus, (children_1, children_2, \cdots, children_n) >, \tag{3.3.6}$$

where the symbol,$\bigoplus$, represnts an OR operator that will act on the n child branches, each of which relate to the same component, with this component being known as the component of the summing node. An example summing node is illustrated in Figure 3.3.3, where all the child event nodes relate to component C.

To construct an Implicant Tree from the BDD node, $G = ite < x, G1, G0 >$, first of all, the failure state, x, and the success state, $\bar{x}$, are identified. Then, the Implicant Trees relating to the failure and success branches of G are obtained, there being $IT(G1)$ and $IT(G0)$ respectively. Next, the logic operations of x AND $IT(G1)$ and $\bar{x}$ AND $IT(G0)$ are computed. Finally, an OR operation is computed for the two new nodes to produce the Implicant Tree node for G. The steps involved in this Implicant Tree construction process are detailed below:

1. If the BDD node is a terminal 1, its Implicant Tree is: IT(1) = 1; if the BDD node is a terminal 0, then the Implicant tree is represented by an NULL node. Otherwise, the BDD node is an *ite* structure and the Implicant Tree is formed according to the following five steps.

2. Obtain the Implicant Tree nodes for the 1-branch, $IT(G1)$, and for the 0-branch, $IT(G0)$ of the BDD separately. This is a recursive process as Implicant Tree will be calculated along each branch until terminal nodes are reached.

3. Identify the variables which will later combine with the Implicant Tree nodes repre-

senting the failure and success branches. For example, for BDD structure shown in Figure 3.2.2:

(a) the variable to combine with the IT node for the 1-branch is x.

(b) the variable to combine with the IT node for the 0-branch is $\bar{x}$.

4. Obtain a new set of Implicant Tree nodes by combining the nodes from step 2 and their corresponding variables from step 3. Suppose the variable from step 3 is x and the IT node for the BDD branch is $IT(G1) = 1$, $IT(G1) = NULL$, or $IT(G1) = IT < y, children >$. The combination rules are described below:

   (a) If $IT(G1) = 1$, then create an IT node, with x being the event and 1 being the child; the Implicant Tree node is denoted as $IT < x, 1 >$; if $IT(G1) = NULL$, then no Implicant Tree node is created. Otherwise, continue with 4.(b) and 4.(c).

   (b) If $IT(G1)$ and x relate to the same component, then (c)i. is used to solve the dependency between the two; otherwise (c)ii. is used (this is always applied for the Implicant Tree obtained from the 1-branch of the BDD, due to the Forward-BDD construction rules).

   (c) If $IT(G1)$ is a sum node, and is represented by

$$IT(G1) = IT < \bigoplus, (children_1, chidren_2, \cdots, children_n) > .$$

   and supposing the i^{th} children node is represented by

$$children_i = IT < y_i, \text{children of } children_i >,$$

      i. This step results in n Implicant Tree nodes. For i from 1 to n, identify the AND operation of x and y_i using the algebra in Equation (3.3.3) and Equation (3.3.4). If $x \cdot y_i = y_i$, then the i^{th} Implicant Tree node is just $children_i$; otherwise, an IT node is created with $x \cdot z$ being the event and the children of $children_i$ being the children, i.e., $IT < x \cdot z, \text{children of } children_i >$.

      ii. Create a new Implicant Tree node with x being the event and $IT(G1)$ being the child node, i.e., $IT < x, IT(G1) >$.

5. If only one Implicant Tree node is formed in step 4., then step 5. is skipped. Oth-

erwise, a sum node is formed with each of the IT nodes derived in step 3. being the children. (If a child node of a sum node is a sum node, then it is equivalent to treating the children of the child sum node as the children of the sum node, as shown in Figure 3.3.4.)

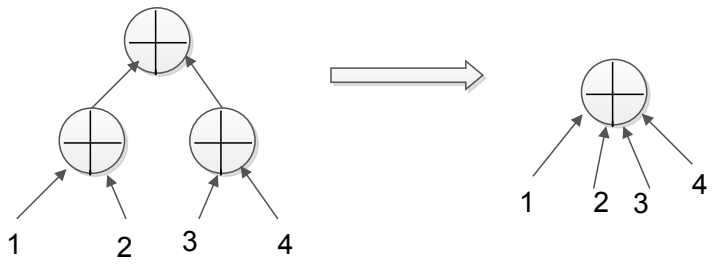

Figure 3.3.4: Merging sum node

6. Store the nodes obtained in step 5 so that the the same Implicant Tree nodes can be reused in cases where the BDD nodes have multiple parents.

3.3.2.3 Quantitative Analysis of Implicant Tree

After the Implicant Tree for a BDD has been obtained, its quantitative analysis can be quickly performed since all variables on a path of the Implicant Tree are independent of each other. The quantification rule for an IT node is given in Equation (3.3.7).

$$P(IT(G)) = \begin{cases} p(x) \cdot P(child) & \text{if } IT(G) \text{ is an event node} \\ P(children_1) + P(children_2) + \cdots + P(children_n) & \text{if } IT(G) \text{ is an sum node} \end{cases} \tag{3.3.7}$$

The probability of a terminal 1 is 1, and during the quantification process, the probability of each Implicant Tree structure is cached for future use.

For the BDD representing the conditions for mission failure, $F_1 + F_2$, shown in Figure 3.3.1, constructed using the Forward-BDD model, the Implicant Tree is constructed as illustrated in Figure 3.3.5. The BDD node $N17$ is converted into Implicant Tree node $IT(N17)$, using the recursive process, presented in Section 3.3.2.2.

The failure probability of $N17$ is calculated by quantifying the Implicant Tree $IT(N17)$:

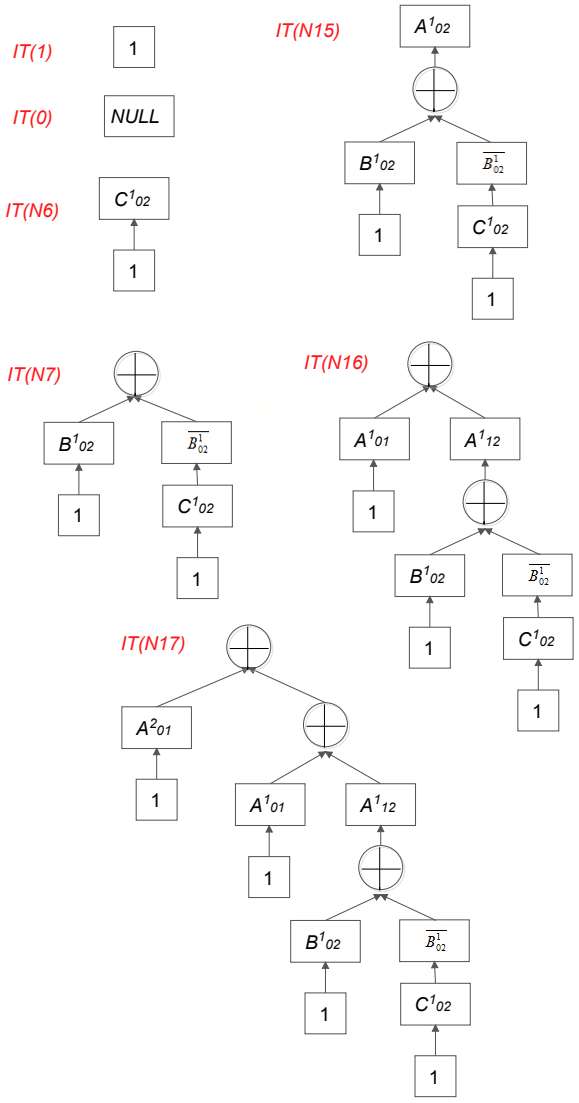

Figure 3.3.5: Implicant Trees for the F-BDD nodes in Figure 3.3.1

$$
\begin{aligned}
P(IT(N6)) =& P(IT < C_{02}^1, 1 >) = p(C_{02}^1) \\
P(IT(N7)) =& P(IT < \bigoplus, (IT < B_{02}^1, 1 >, IT < \overline{B_{02}^1}, IT(N6) >) >) \\
=& P(IT < B_{02}^1, 1 >) + P(IT < \overline{B_{02}^1}, IT(N6) >) \\
=& p(B_{02}^1) + p(\overline{B_{02}^1}) \cdot P(IT(N6)) \\
P(IT(N15)) =& P(IT < \bigoplus, (IT < A_{02}^1, IT(N7) >, IT < A_{01}^2, 1 >) >) \\
=& P(IT < A_{02}^1, IT(N7) >) + P(IT < A_{01}^2, 1 >) \\
=& p(A_{02}^1) \cdot P(IT(N7)) + p(A_{01}^2) \\
P(IT(N17)) =& P(IT < \bigoplus, (IT < A_{01}^1, 1 >, IT < A_{12}^1, IT(N7) >, IT < A_{01}^2, 1 >) >) \\
=& P(IT < A_{01}^1, 1 >) + P(IT < A_{12}^1, IT(N7) >) + P(IT < A_{01}^2, 1 >) \\
=& p(A_{01}^1) + p(A_{12}^1) \cdot P(IT(N7)) + p(A_{01}^2) \\
=& p(A_{01}^1) + p(A_{01}^2) + [p(A_{02}^1) - p(A_{01}^1)] \cdot [p(B_{02}^1) + p(C_{02}^1) - p(B_{02}^1) \cdot p(C_{02}^1)]
\end{aligned}
\tag{3.3.8}
$$

Therefore,

$$
Q_{miss} = p(A_{01}^1) + p(A_{01}^2) + [p(A_{02}^1) - p(A_{01}^1)] \cdot [p(B_{02}^1) + p(C_{02}^1) - p(B_{02}^1) \cdot p(C_{02}^1)] \tag{3.3.9}
$$

This is the same result as those obtained in Equation (3.2.3), Equation (3.2.7), and Equation (3.2.9).

3.3.3 Improvement to the Forward-BDD Model

Although the quantitative analysis of the Implicant Tree is efficient, the construction of the Implicant Tree data structure could be time consuming for large BDDs. This would not be a desirable feature for a phased mission analysis technique that would be used as part of a decision making process. Therefore, a new quantification method is developed to evaluate Forward-BDDs directly and the efficiency of the proposed method is demonstrated in the last section of this chapter. The use of the Forward-BDD together with the new quantification method is denoted as Model 3.

3.3.3.1 New Quantitative Rules

In Model 3, the 1-branch always links two variables that relate to different components and the 0-branch can either link two variables that relate to different components or two variables that relate to the same component because of the rules used to construct Forward-BDDs. The quantification method proposed is based on the phase algebra in Equation (3.1.1) and dependence algebra in Equation (3.1.2).

For a BDD node as shown in Figure 3.2.2, $G = ite < x, G1, G0 >$, where $G1 = ite < y, H1, H0 >$ and $G0 = ite < z, I1, I0 >$, variables x and y always relate to different components. The quantification rules consider the relationship between x and z is given by Equation (3.3.10):

$$P(G) = \begin{cases} p(x)P(G1) + [1 - p(x)]P(G0) & cp(x) \neq cp(z) \\ P(G0) + p(x)[P(G1) - P(I1)] & cp(x) = cp(z), fm(x) = fm(z) \\ P(G0) + p(x)[P(G1) - P(I0^*)] & cp(x) = cp(z), fm(x) \neq fm(z) \end{cases} \quad (3.3.10)$$

where

- $I0^* = (I0)_{x=1}$, is the first node with variable relating to a component other than x encountered during the traversal down the 0-branch of node $G0$.

3.3.3.2 Proof of the Quantification Rules

Since $cp(x) \neq cp(y)$ in all cases, $P(x \cdot G1) = p(x)P(G1)$.Three cases are then carried. When $cp(x) \neq cp(z)$:

Proof 3.3.1

$$\begin{aligned} P(G) &= P(x \cdot G1 + \bar{x} \cdot G0) \\ &= P(x \cdot G1) + P(\bar{x} \cdot G0) \\ &= p(x) \cdot P(G1) + p(\bar{x}) \cdot P(G0) \\ &= p(x) \cdot P(G1) + (1 - p(x)) \cdot P(G0) \end{aligned}$$

When $cp(x) = cp(z)$ and $fm(x) = fm(z)$, the phase algebra given in Equation (3.1.2) are used for the following deviation.

Proof 3.3.2

$$
\begin{aligned}
P(\bar{x}\cdot G0) &= P(\bar{x}\cdot(z\cdot I1+\bar{z}\cdot I0))\\
&= P(\bar{x}\cdot z\cdot I1+\bar{x}\cdot\bar{z}\cdot I0)\\
&\textit{(according to the phase algebra, } \bar{x}\cdot z = z - x\cdot z \textit{ and } \bar{x}\cdot\bar{z}=\bar{z}\textit{)}\\
&= P((z-x\cdot z)I1+\bar{z}\cdot I0)\\
&= P(z\cdot I1+\bar{z}\cdot I0-x\cdot zI1)\textit{(since } x\cdot z = x\ \textit{)}\\
&= P(G0-x\cdot I1)\\
&= P(G0)-p(x)\cdot P(I1)
\end{aligned}
$$

Here, $(I1)_{x=1} = I1$ *always holds because the variable of* $I1$ *is different to* x*. Substituting* $P(\bar{x}\cdot G0)$ *into* $P(G)$ *gives:*

$$
\begin{aligned}
P(G) &= P(x\cdot G1+\bar{x}\cdot G0)\\
&= P(x\cdot G1)+P(\bar{x}\cdot G0)\\
&= p(x)\cdot P(G1)+P(G0)-p(x)\cdot P(I1)\\
&= P(G0)+p(x)\cdot[P(G1)-P(I1)]
\end{aligned}
$$

When $cp(x) = cp(z)$ and $fm(x) \neq fm(y)$, the dependence algebra given in Equation (3.1.2) are used in the following derivation.

Proof 3.3.3

$$
\begin{aligned}
P(\bar{x}\cdot G0) &= P(\bar{x}\cdot(z\cdot I1+\bar{z}\cdot I0))\\
&= P(\bar{x}\cdot z\cdot I1+\bar{x}\cdot\bar{z}\cdot I0)\\
&\textit{(according to dependence algebra, } \bar{x}\cdot z = z \textit{ and } \bar{x}\cdot\bar{z}=\bar{z}-x\cdot\bar{z}\textit{)}\\
&= P(z\cdot I1+\bar{z}\cdot I0-x\cdot\bar{z}\cdot I0)\\
&\textit{(since } x\cdot\bar{z}=x\textit{, according to Equation } (3.1.2)\textit{)}\\
&= P(G0-x\cdot I0)\\
&= P(G0)-p(x)\cdot p(I0^{*})
\end{aligned}
$$

Here, $I0^* = (I0)_{x=1}$. *Substituting* $P(\bar{x} \cdot G0)$ *into* $P(G)$ *gives:*

$$\begin{aligned} P(G) &= P(x \cdot G1 + \bar{x} \cdot G0) \\ &= P(x \cdot G1) + P(\bar{x} \cdot G0) \\ &= p(x) \cdot P(G1) + P(G0) - p(x) \cdot P(I0^*) \\ &= P(G0) + p(x) \cdot (P(G1) - P(I0^*)) \end{aligned}$$

The proposed quantification method can be used to quantify the Forward-BDDs directly rather than requiring another data structure to be constructed prior to quantification. Therefore, the quantification efficiency is expected to be improved greatly compared with the Implicant Tree method. The testing and comparison of the methods are shown in the last section of this chapter.

3.3.3.3 Example

The Quantification of the BDD shown in Figure 3.3.1 using the proposed quantification method is given below. Using the case when $cp(x) = cp(z)$ and $fm(x) \neq fm(z)$ from Equation (3.3.10):

$$\begin{aligned} P(N17) &= P(G0) + p(x) \cdot [p(G1) - P(I0^*)] \\ &= P(N16) + p(A^2_{01})[1 - 0] \\ &= P(N16) + p(A^2_{01}) \end{aligned} \tag{3.3.11}$$

The probability of $N16$ is calculated using the case when $cp(x) = cp(z)$ and $fm(x) = fm(z)$ from Equation (3.3.10):

$$\begin{aligned} P(N16) &= P(G0) + p(x) \cdot [p(G1) - P(I1)] \\ &= P(N15) + p(A^1_{01})[1 - P(N7)] \\ &= p(A^1_{02})P(N7) + p(A^1_{02}) - p(A^1_{01}) \cdot P(N7) \end{aligned} \tag{3.3.12}$$

Substituting $P(N16)$ into $P(N17)$ gives:

$$P(N17) = p(A^1_{01}) + p(A^2_{01}) + [p(A^1_{02}) - p(A^1_{01})] \cdot [p(B^1_{02}) + p(C^1_{02}) - p(B^1_{02}) \cdot p(C^1_{02})] \tag{3.3.13}$$

Mission unreliability obtained by the Implicant Tree method and the proposed quantification method are the same, while the proposed quantification method requires less

computation effort as it does not need to construct an Implicant Tree as in Figure 3.3.5 and quantification process is much more simpler.

This result is identical to that obtained using the other methods considered in this chapter. However, the Implicant Tree method of the Forward-BDD requires less computational effort than the Implicant Tree method since there is no need to construct the Implicant Tree and hence the quantification process is simpler.

3.4 The Comparison and Conclusion

In this chapter, two existing BDD models for the analysis of PMSs containing components with multiple failure modes were reviewed, the DEP-BDD model and the Forward-BDD model. Since the DEP-BDD model has been shown to be inaccurate [35], two amendments are proposed to correct the errors in the DEP-BDD analysis and hence to obtain correct results. The two amendments lead to two new models, Model 1 and Model 2. The Forward-BDD model analyses BDDs by first converting the BDD to an Implicant Tree. This extra layer of analysis might be expected to lead to inefficiency and therefore a new quantification method is proposed to replace the Implicant Tree method in the Forward-BDD model. This analysis is denoted as Model 3.

3.4.1 Testing Results

The accuracy of the DEP-BDD model and the Forward-BDD model were compared in [36] and it was confirmed that the Forward-BDD generates exact individual phase and mission unreliabilities. Since the failure probabilities obtained using the three new models are identical to those obtained using the Forward-BDD model, the four models are compared here solely in terms of the efficiency of their analysis. Two efficiency measures are used to assess the performance of the models: the size of the BDD representing mission failure, which is defined as the number of nodes in the BDD and the analysis time for the mission as a whole.

To compare the performance of the developed ordering schemes and BDD models, a large number of PMSs with different features needs to be tested.

A programme was written to generate reliable benchmark fault trees with varying sizes and diverse structure features using the method presented in [24]: the maximum number of components (varying from 20 to 80), the maximum number of inputs in each fault tree layer (varying from 6 to 16), the percentage of gate inputs in each fault tree layer, a/b,

where a is the percentage for the first three layers, and b is the percentage for the other layers (a varies from 0.7 to 0.9, and b varies from 0.1 to 0.3), the maximum number of component failure modes (varying from 2 to 4), and the percentage of multiple failure mode components (varying from 0.1 to 0.6). By inputting the selected parameter values to the algorithm presented in Appendix A, fault trees are generated.

In the general case, the more components that are involved and the more inputs on each layer, the bigger the fault tree; the more failure modes and phases involved in a mission, the more phase and failure mode dependencies must be addressed. Thus, the most complex fault tree is generated using the maximal value of each parameter.

The fault trees obtained using this method are equivalent to fault trees that have undergone restructuring and modularisation so that the effects of these factors on analysis speed need not be counted. The ranges of the parameter values are chosen to produce reasonable size fault trees to solve and illustrate the comparison of the BDD models.

A computer with an Intel(R) Core 3.10GHz processor and 4 GB RAM is used. The models are constructed and analysed in a bespoke C++ code and analysis time is computed using the clock_t() function and presented in units of seconds.

3.4.1.1 Model Efficiency

The efficiency analysis of Model 1, 2, and 3 and the Forward-BDD model is compared by examining the sizes of the BDDs constructed for each model and the time taken to perform a quantitative analysis.

In appendix, Table B.1.1 shows the sizes of the BDD representing mission failure constructed for each model (in column 2 to 5) and the time taken to quantify the mission unreliability in each case (in column 6 to 9).

Table 3.4.1 summarises the results relating to size of the BDD by showing the number of missions for which each model produces a BDD smaller than that produced by the other models. Model 3 shares its BDD construction rules with the Forward-BDD model and therefore shares its results. These models are seen to generate smaller BDDs than the other two models for the majority of the PMS considered.

Table 3.4.2 shows the number of missions for which the analysis of each model takes less time than the analysis of the same mission using other models. It illustrates that all of the developed BDD models take less analysis time than the Forward-BDD model and therefore improved efficiency can be expected when using these models. Within the three

developed BDD models, Model 3 can be seen to generally perform better than Model 1 and Model 2, since it leads to the smallest analysis time for around 50% of the missions.

Table 3.4.3 gives the efficiency improvement of the three developed BDD models over the Forward-BDD model computed by the average reduced mission analysis time. All of the new BDD models show greatly reduced analysis time in comparison to the Forward-BDD model. A particularly noteworthy result is seen for Model 3, which, despite using the same BDD structure as the Forward BDD model, results in an analysis time that is an average 90.09%, less than that required for the Forward BDD model.

BDD model	Model 1	Model 2	Model 3 and Forward-BDD
No. of missions	23	18	59

Table 3.4.1: Performance comparison of BDD models in terms of BDD sizes

BDD model	Model 1	Model 2	Model 3	Forward-BDD
No. of missions	30	19	51	0

Table 3.4.2: Performance comparison of BDD models in terms of analysis time

BDD models	Model 1	Model 2	Model 3
Average improvement	75.28%	54.12%	90.09%

Table 3.4.3: Analysis of the efficiency improvement (percentage of time reduced) of the three developed BDD models over the Forward-BDD model

3.4.1.2 Comparison of the Three New Models

Table 3.4.2 and Table 3.4.3 shows that all three new models are more efficient than the Forward-BDD model, as they require much less mission unreliability analysis time. This would make them more desirable for use as part of a decision support tool for PMS than the Forward-BDD model, Table 3.4.4 and Table 3.4.5 compare the three models according to the BDD size and mission analysis time.

Table 3.4.4 gives the average reduction in the size of the BDD representing mission failure of each model over the other two models, individually and on average. The more the BDD size is reduced over the other models, the more efficient the model is in terms of its construction. The average reduction in size of BDDs constructed using Model 1 compared to the size of the BDDs constructed using Model 2 and Model 3 is negative, as is the average reduction in size of the BDDs constructed using Model 2 compared to Model 1 and Model 3. This demonstrates that the BDD construction method of Model 3 and the

Forward-BDD model lead to smaller BDD sizes in general and is therefore more efficient than the BDD construction methods used in Model 1 and Model 2. These results also support the conclusion in [36], that for all tested mission configurations, the Forward-BDD model (or Model 3) resulted in smaller BDDs.

BDD model	Model 1 over		Model 2 over		Model 3 over	
	Model 2	Model 3	Model 1	Model 3	Model 1	Model 2
Average reduction in BDD size	-2.37%	-28.41%	-19.84%	-48.10%	10.59%	13.87%
Average reduction in BDD size over other models	-15.39%		-33.97%		12.23%	

Table 3.4.4: Average percentage reduction in BDD sizes compared to the other models

The comparison of BDD model efficiency measured by the mission unreliability analysis time is presented in Table 3.4.5. The table presents the average percentage of reduction in analysis time of one model when compared to the other two models individually and on average. The average reduction in time using Model 3 instead of the other two models is 30.35%, which demonstrates the average of analysis efficiency that Model 3 has over the other two models. These results suggests that Model 3 is most likely to give the best performance of the three models when calculating the overall mission unreliability and might therefore be the first choice when choosing which method to use when the time taken to perform an analysis is important.

BDD models	Model 1 over		Model 2 over		Model 3 over	
	Model 2	Model 3	Model 1	Model 3	Model 1	Model 2
Average reduction in analysis time	11.19%	3.28%	8.55%	-2.47%	27.84%	32.85%
Average reduction in analysis time over other models	7.24%		3.04%		30.35%	

Table 3.4.5: Average percentage reduction in analysis time compared to the other models

3.4.2 Summary

If a reliability analysis methodology is to be used to support real-time decision making for systems operating phased missions in changing mission environments, it is crucial that the applied methodology can analyse PMS quickly and accurately.

Dependencies that arise due to the phases in a mission and components that have multiple failure modes increase the complexity of reliability analysis in comparison to the analysis of systems with a single phase of operation and components with a single failure mode. There are two existing BDD models, which account for these dependencies during BDD construction: the DEP-BDD model [43]and the Forward-BDD model [36][35]. The DEP-BDD model constructs BDDs by applying phase algebra, Equation (3.1.1) and dependency algebra, Equation (3.1.2), to account for the dependencies that

arise due to mission phases and components with multiple failure modes. The quantitative analysis of the DEP-BDD model can be performed quickly but can give inaccurate results [35][36]. Instead of correcting the model, the researchers developed an alternative model, the Forward-BDD model, which uses a forward phase level variable ordering to construct BDDs and the Implicant Tree method for quantification.

In this Chapter, after reviewing the DEP-BDD model, two amendments, Model 1 and Model 2, have been proposed, which correct the previously-observed inaccuracies. For the Forward-BDD model, the quantification process requires construction of another data structure, the Implicant Tree, before evaluation. This can be quite time consuming and is therefore not conducive to fast analysis. A more efficient quantification method is developed using the dependency algebra presented in [43].

Since Model 1, Model 2, Model 3 and the Forward-BDD model generate accurate unreliabilities, the last section compared the analysis efficiency of the four models by testing them on a large number of randomly-generated phased missions. The results show all of the three developed models offer much faster analysis for the PMS with multiple failure modes, than the Forward-BDD model. The inefficiency of the Forward-BDD model is due to the fact that before quantification can take place, the constructed BDDs need to be converted to Implicant Trees in order to address the dependencies between variables. Model 3 has the same BDD structure as the Forward-BDD model, therefore, the analysis efficiency advantage of Model 3 over the Forward-BDD is purely down to the improvement in efficiency that comes from the proposed quantification method.

None of the models is capable of generating the smallest BDD in all cases but Model 3 (or the Forward-BDD model) was shown to have a higher chance of obtaining smaller BDDs compared with the two models that are based on DEP-BDD analysis. This result is in accordance with the conclusion stated in [36] that the Forward-BDD resulted in smaller BDDs. Of the three new models, Model 3 was seen to result in the highest percentage reduction in mission analysis time when compared to the other two models for the mission configurations tested, meaning that Model 3 would appear to be the most promising to be used when performing reliability analysis for PMS containing components with multiple failure modes to help the system make real-time decisions as to its next course of action in dynamic, rapidly changing mission environments.

Chapter 4

Variable Ordering Schemes for PMS with Multiple Failure Mode Components

4.1 Introduction

BDD construction initially requires the variables to be ordered and the variable ordering can have a big impact on the size of the BDD [11]. A non-optimal ordering scheme can result in an exponential increase in BDD size. Since the size of a BDD has a direct impact on the time it tasks to perform quantitative analysis, a good choice of ordering scheme can lead to faster analysis. Therefore, if the reliability analysis of a PMS is to be used to support a decision making process, the ordering scheme used to order variables within the BDD can directly affect how quickly decisions can be made as to the best next course of action. If the mission configuration changes, variables may need to be reordered while the mission is in process to account for all variables related to the next mission tasks and ensure the sizes of the BDDs required to analyse the new mission configuration are as small as possible. However, the variable reordering and repeating analysis of previous phases will take time and slow down the decision making process. A more efficient scheme which can avoid such repetitious work needs to be developed to allow faster unreliability computation of the following phases of the updated mission configuration in order to enable fast decisions to be made as to the best next course of action.

Though some work has been carried out to investigate variable ordering schemes for standard fault trees with a single phase and single failure modes [37][10][9][8][15][16][21], no

research has focused on investigating the effects of variables ordering on PMS whose failure is affected by components that fail in multiple failure modes. In this section, nine variable ordering schemes are investigated; eight of these are extension of ordering schemes applied to standard, non-phased mission fault trees and the other one is a novel scheme, which is developed specifically for application within a decision making process presented in Section 2.6 and whose aim is to allow the fastest possible calculation of updated unreliability when required within that process.

Variables in a PMS contains information about which component it relates to, in which failure mode it fails and in which phase it fails. Therefore, variables must be ordered in three levels [22]: at component level, phase level and failure mode level. The component level ordering is the most complex aspect of variable ordering since the number of components can be enormous in PMS. There are two types of phase ordering, forward ordering considers variables in order of their phase index and backward ordering considers variables according to reverse phase index [45]. The only requirement for failure mode ordering is the ordering of failure modes in all phases is consistent.

The variable ordering scheme investigation in this research focuses on component level ordering, since the number of ways of ordering components can grow exponentially with the number of components in the PMS, and those leading to compact BDD sizes are preferred. In this chapter, eight ordering schemes designed for standard fault trees with single phase and single failure mode components are extended to allow ordering of variables in PMS containing components that fail in multiple failure modes. A new ordering scheme is then proposed, in which all variables are ordered before the mission starts and can be applied no matter how the mission configuration changes. All nine component ordering schemes are tested on a set of phased missions and the results are used to compare the effects of the different schemes in the last section.

4.2 Extension of Standard Ordering Schemes

For fault trees that do not model phased missions and consider only single component failure modes, variables only relate to a component; no other information must be encoded by the variables. However, for the PMS considered in this research, variables also hold information about the phase in which a component failure occurs and the component failure mode. This means existing variable ordering schemes from the literature must be amended in order to allow then to be applied to the PMS under consideration.

Eight ordering schemes for standard fault trees are listed below.

1. Modified Top-down Ordering Scheme [37].
2. Modified Depth-first Ordering Scheme [10].
3. Modified Priority Depth-first Ordering Scheme [37].
4. Modified Leaves Depth-first Ordering Scheme[10][15][16].
5. Non-Dynamic Top-down Weighted Ordering Scheme [37].
6. Dynamic Top-down Weighted Ordering Scheme [37].
7. Bottom-up Weighted Ordering Scheme [10][21].
8. Event Criticality Ordering Scheme [9][8].

In order to allow the principles of these existing ordering schemes to be applied to the phased mission considered in this work, the fault tree representing overall mission failure must be taken into account. Ordering schemes 1 to 7 are then applied by considering this mission failure fault tree whilst neglecting the phase and failure mode indices [22]. This version of a fault tree will be referred to as a 'don't care' fault tree.

Variables can then be sorted according to the principles of the existing original ordering schemes since the variables in the 'don't care' fault tree only indicate which component it relates to. The 8^{th} scheme is applied to the 'don't care phase' fault tree, where only the phase indices associated with the variables are eliminated. The final variable ordering is then defined by applying the rules that are developed in the following sections.

For all of the schemes, event inputs are considered before gate inputs and if events or gates cannot be sorted using the principles of the schemes, those that appear earlier, i.e. towards the top left of the fault tree, are given priority.

4.2.1 Example System

Consider a non-repairable UAV that can perform 4 different tasks: task-1, task-2, task-3, and task-4. The fault trees representing the failure logic of each task are shown in Figure 4.2.1. There are 6 components: A, B, C, D, E, and F. Components A, C, E and F can fail in two failure modes; components B and D can fail in three failure modes. For example, A^2 represents component A failing in failure mode 2. In mission M_A, the UAV is required to perform task-1, task-2 and task-3 in sequence. Consider also that the UAV

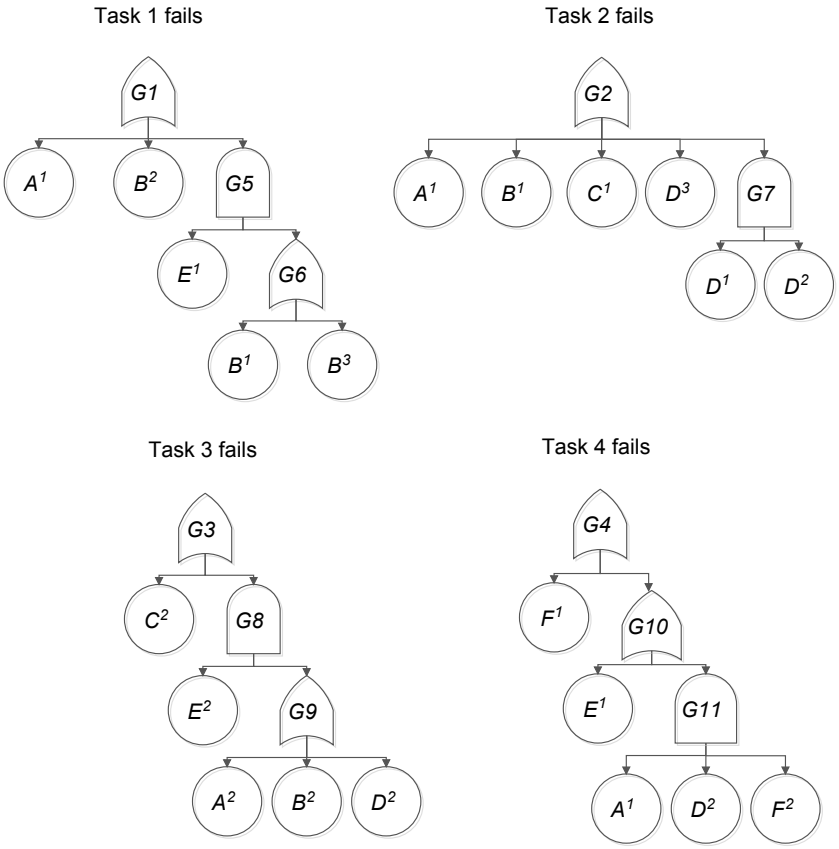

Figure 4.2.1: Fault trees for failure of tasks the example PMS can perform

must autonomously make decisions as to whether to carry on with mission M_A after an internal failure is detected and if not, find an optimal alternative in a timely manner. A methodology that allows the UAV to quickly quantify the probability of success for each of the possible missions is required in order to decide which mission to perform. The chosen mission will be the one with the highest probability of success. The failure of mission M_A is represented by the OR combination of failures in all phases as shown in Figure 4.2.2.

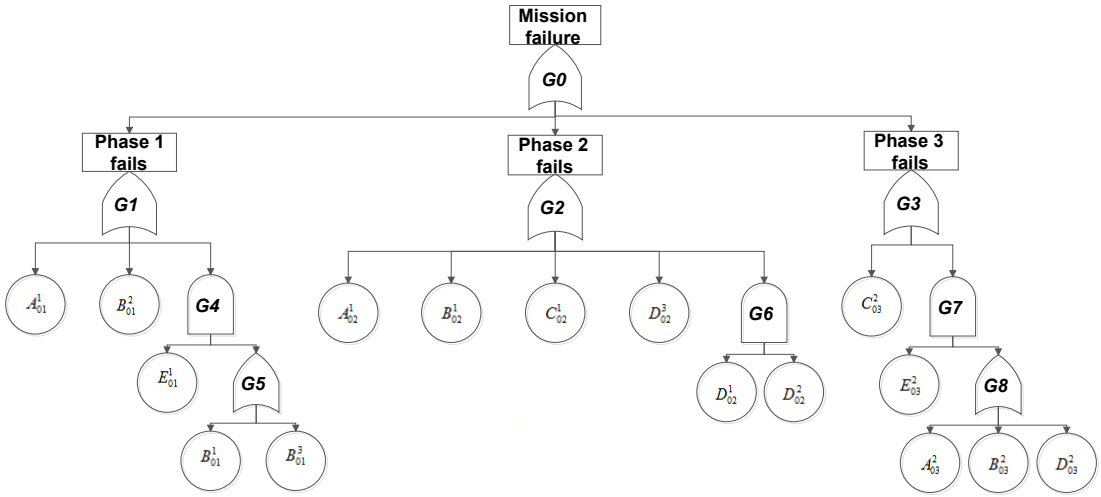

Figure 4.2.2: The fault tree representing mission M_A failure for the example system

The 'don't care' fault tree for the failure of M_A is shown in Figure 4.2.3. Schemes 1 to 7 order variables of the PMS using Figure 4.2.3.

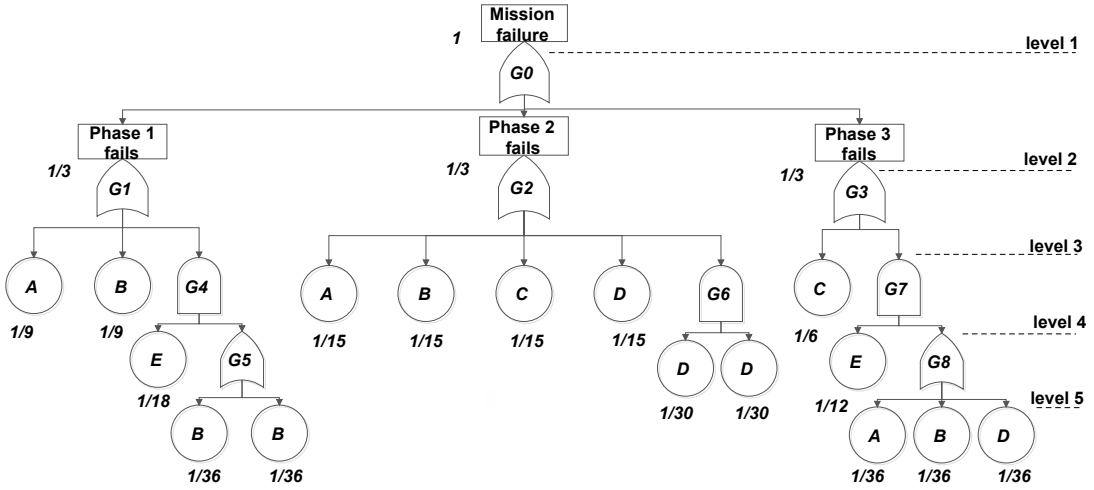

Figure 4.2.3: The 'don't care' fault tree for M_A

4.2.2 Modified Top-down Ordering (Scheme 1)

Scheme 1 orders variables as they appear in the 'don't care' fault tree in a top-down, left-right arrangement. Variables that appear higher in the tree (smaller level number) are listed earlier. For tied variables, the one that occurs the most is allocated earlier.

Consider the 'don't care' fault tree for M_A, as shown in Figure 4.2.3. The variables that appear in level 3 are: A, B, C, and D; component E appears only in level 4. Note that, the smaller the level number, the higher the level as illustrated in Figure 4.2.3. Therefore, the components appearing in level 3 are ordered before E. These four variables, are ordered according to their occurrence number. A occurs 3 times, B occurs 5 times, C occurs 2 times, and D occurs 4 times. Therefore, by descending occurrence number, they are ordered as: $B < D < A < C$. All parameters required to arrange the variables are listed in Table 4.2.1:

Component	First level of appearance	Occurrence Number	Appearance sequence
A	3	3	1
B	3	5	2
C	3	2	4
D	3	4	5
E	4	2	3

Table 4.2.1: Parameters of components for Scheme 1

The variables are ordered by increasing first level of appearance, decreasing occurrence number and increasing appearance sequence. Therefore, the order list according to Scheme 1 is:

$$B < D < A < C < E. \tag{4.2.1}$$

4.2.3 Modified Depth-first Ordering (Scheme 2)

Scheme 2 considers the 'don't care' fault tree in a modified depth-first manner. It treats the fault tree as being made up of small sub-trees, and each sub-tree is fully explored according to Scheme 1 before the next sub-tree is considered. The gates are considered from left to right as they appear in the mission failure fault tree. Therefore, the left most gate is explored before considering any of the remaining gates.

Using Scheme 2 to order variables in the 'don't care' fault tree in Figure 4.2.3, the gate inputs $G1$, $G2$ and $G3$ of the top gate $G0$ are considered in sequence and are analysed

according to the principles of Scheme 1 individually. For $G1$, Scheme 1 gives order list: $B < A < E$; for $G4$, since components A and B, are already allocated, only D and C need to be arranged. Scheme 1 gives the order: $D < C$. There are added after the last element in the order list of the last visited gate $G1$, component E. The order list obtained is: $B < A < E < D < C$. All components have now been ordered, and therefore gate $G6$ does not need to be explored any more. The final variable order list according to Scheme 2 is:

$$B < A < E < D < C. \tag{4.2.2}$$

4.2.4 Modified Priority Depth-first Ordering (Scheme 3)

Scheme 3 is a modified version of Scheme 2. Using Scheme 3, the gate inputs (sub-trees) in a 'don't care' fault tree are also explored using Scheme 1 and one gate input is fully explored before analysing the next one. The difference from Scheme 2 is that, gate inputs are considered in a specified order rather than simply from left to right. Gates with only event inputs are considered before those with both event and gate inputs.

Applying Scheme 3 to the 'don't care' fault tree in Figure 4.2.3 to order variables, unlike Scheme 2, which considers gate inputs as they appear from left to right, Scheme 3 gives priorities to gate inputs that have only event inputs. In this example, no gate inputs of $G0$ contain only event inputs, thus $G1$, $G2$ and $G3$ are considered from left to right as they appear in the tree and are explored according to the Scheme 1 principles one-by-one. Since gates $G1$, $G2$, and $G3$ have only one gate input and so do their sub-trees if any, Scheme 3 gives the same variable order list as Scheme 2:

$$B < A < E < D < C. \tag{4.2.3}$$

4.2.5 Modified Leaves Depth-first Ordering (Scheme 4)

Scheme 4 is also a modified version of Scheme 2. Variables under one gate are all allocated before ordering those under the next gate. A gate is considered first if it:

1. Contains the smallest number of leaves, which is defined as the total number of basic events beneath the gate. In case of ties, then the gate will be considered first if it
2. Contains the smallest number of unconsidered leaves.

Still, variables that occur most frequently are listed earlier.

Scheme 4 is now used to order variables in the 'don't care' fault tree in Figure 4.2.3. Gate inputs $G1$ and $G3$ contain 5 leaves while $G2$ contains 6 leaves, and $G1$ appears before $G3$. Thus, gates are considered in the sequence of $G1 < G3 < G2$. The parameters used to decide the sequence of gates to be considered are shown in Table 4.2.2.

Gate	Leaves No.	Unordered leaves No.	Appearance sequence
G1	5	5	1
G2	6	6	2
G3	5	5	3

Table 4.2.2: Parameters of gate inputs of $G0$ for Scheme 4

Exploring $G1$ using Scheme 1 with sub-gates (if they exists) considered following the principles described in Scheme 4 gives: $B < A < E$. Exploring $G3$ gives: $C < D$. Adding the order list of $G3$ to that of $G1$ gives the full variable order list:

$$B < A < E < C < D. \tag{4.2.4}$$

Since all variables are arranged, there is no need to explore the remaining gate $G2$.

4.2.6 Non-Dynamic Top-down Weighted Ordering (Scheme 5)

Scheme 5 lists a variable in the 'don't care' fault tree before others if it has:

1. The smallest contribution weight to the top event of the 'don't care' fault tree. The weight is calculated as follows: the top event is given a weight 1; working down through the fault tree, the weight of each gate is equally distributed between its inputs. Weights of repeated variables are added together. In case of ties, then a variables is ordered earlier if it has:

2. The highest average level of appearance, which is calculated by the sum of the levels on which the variable appears, divided by how many times it occurs. In case of ties, then a variables is ordered earlier if it has:

3. The highest number of occurrences. In case of ties, then a variables is ordered earlier if it has:

4. The highest priority in the order list when applying Scheme 1.

Scheme 5 is now used to order variables in the 'don't care' fault tree in Figure 4.2.3. The weight of each variable is calculated by giving weight 1 to the top event and the

weight at each gate is equally distributed between the inputs of the gate. For example, there are three gate inputs of $G0$, whose weight is 1, and therefore, each gate input, $G1$, $G2$ and $G3$, is assigned a weight of 1/3. The weight of each variable appears in the fault tree is shown in Figure 4.2.3.

The weights of the same variable are added to achieve the weight for that variable:

$$\begin{aligned} Weight(A) &= \frac{1}{9} + \frac{1}{15} + \frac{1}{36} = \frac{37}{180} \\ Weight(B) &= \frac{1}{9} + \frac{1}{36} + \frac{1}{36} + \frac{1}{15} + \frac{1}{36} = \frac{47}{180} \\ Weight(C) &= \frac{1}{6} + \frac{1}{15} = \frac{42}{180} \\ Weight(D) &= \frac{1}{15} + \frac{1}{30} + \frac{1}{30} + \frac{1}{36} + = \frac{29}{180} \\ Weight(E) &= \frac{1}{18} + \frac{1}{12} = \frac{25}{180}. \end{aligned} \tag{4.2.5}$$

By decreasing contribution weight, the variables are ordered as:

$$B < C < A < D < E. \tag{4.2.6}$$

4.2.7 Dynamic Top-down Weighted Ordering (Scheme 6)

Scheme 6 is a scheme that applies Scheme 5 to a changing series of dynamic fault trees. On the one hand, Scheme 6, follows the same ordering principles as Scheme 5; on the other hand, once a variable has been allocated, the variable is deleted from the 'don't care' fault tree wherever it appears. Weights are then reassigned to the modified fault tree to allocate another variable. The process repeats until all variables haven been sorted.

Scheme 6 is now used to order variables in the 'don't care' fault tree in Figure 4.2.3. Component B is allocated first according to Scheme 5. Then, all events relate to B are deleted from the fault tree in Figure 4.2.3 and creating a new fault tree as illustrated in Figure 4.2.4. Scheme 5 is used to order variables in the fault tree in Figure 4.2.4, with weights reassigned to each variable and gate. The total weight for each of the remaining variables is calculated as follows:

$$\begin{aligned} Weight(A) &= \frac{1}{6} + \frac{1}{12} + \frac{1}{24} = \frac{7}{24} \\ Weight(C) &= \frac{1}{6} + \frac{1}{12} = \frac{6}{24} \\ Weight(D) &= \frac{1}{12} + \frac{1}{24} + \frac{1}{24} + \frac{1}{24} = \frac{5}{24} \\ Weight(E) &= \frac{1}{6} + \frac{1}{12} = \frac{6}{24}. \end{aligned} \tag{4.2.7}$$

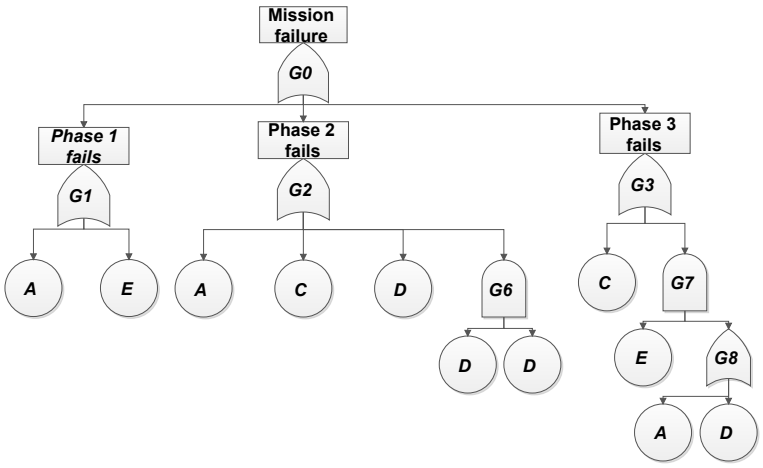

Figure 4.2.4: The new fault tree after B is deleted from the fault tree in Figure 4.2.3

In this fault tree, variable A has the highest weight and thus is allocated after B in the ordered list. All events relating to A appear in the fault tree in Figure 4.2.4 are deleted to form another new fault tree. The process repeats until all variables are sorted. The final order list obtained using Scheme 6 is:

$$B < A < E < D < C. \tag{4.2.8}$$

4.2.8 Bottom-up Weight Ordering (Scheme 7)

Using Scheme 7, gates are explored in a depth-first manner (as in Scheme 2) and a gate is explored first if it has:

1. The highest weight. In case of ties,
2. The highest percentage of repeated variables, which is calculated by dividing the total number of leaves by the number of repeated leaves, e.g. the percentage of repeated variables of $G1$ is $3/5 = 0.6$.

The weight of a gate is calculated by assigning a weight is to each variable and working through the fault tree in a bottom-up manner adding weights together to obtain the weight of each gate according to the logic type and the event inputs of the gate:

- The weight of an AND gate if there are no repeated event inputs: $W_{AND} = \prod_{i=1}^{n} q_i$.
- The weight of an AND gate if there are repeated event inputs: $W_{AND} = 0$.
- The weight of an OR gate: $W_{OR} = 1 - \prod_{i=1}^{n}(1 - q_i)$,

where q_i is the weight of the i^{th} input of the gate. Event inputs have a weight of $q_i = 1/(m+1)$ where m is the maximum number of failure modes in which any event can fail. For gate inputs, q_i is the weight of the gate as defined above.

Scheme 7 is used to order variables in the 'don't care' fault tree in Figure 4.2.3. The maximum number of failure modes of any variable is 3 and thus each event input is assigned with weight 1/4 and the weight of each gate is calculated as follows:

$$\begin{aligned}
&Weight(G5) = 1 - (1 - 0.25)^2 = 0.4375 \\
&Weight(G6) = 0 \\
&Weight(G8) = 1 - (1 - 0.25)^3 = 0.5781 \\
&Weight(G4) = 0.25 * Weight(G5) = 0.1094 \\
&Weight(G7) = 0.25 * Weight(G8) = 0.1445 \\
&Weight(G1) = 1 - (1 - 0.25) * (1 - 0.25) * (1 - Weight(G4)) = 0.4990 \\
&Weight(G2) = 1 - (1 - 0.25)^4 * (1 - weight(G6)) = 0.6836 \\
&Weight(G3) = 1 - (1 - 0.25) * (1 - weight(G7)) = 0.3584.
\end{aligned} \tag{4.2.9}$$

For gate inputs of the top event, $G2$ is explored before $G1$, and $G3$ is considered at last, since $Weight(G2) > Weight(G1) > Weight(G3)$. Those gates are explored in a depth-first manner one after the other until all variables are ordered. Sorting variables under $G2$ using the parameters shown in Table 4.2.1 gives: $B < D < A < C$. Exploring $G1$ gives the final order list:

$$B < D < A < C < E. \tag{4.2.10}$$

All variables are now allocated and there is no need to explore the remaining gate, $G3$.

4.2.9 Event Criticality Ordering (Scheme 8)

Scheme 8 uses Birnbaum's structural importance measure to order variables in the 'don't care phase' fault tree. However, since the 'don't care phase' fault tree includes information relating to the failure modes in which the events fail, the calculation of Birnbaum's importance measure needs to be modified to take multiple failure modes into account. The steps involve firstly calculating the importance measure for each distinct event in the 'don't care phase' fault tree that relates to different failure modes but of the same component and

secondly taking the weighted average of the importance measures of events relate to the same component to obtain an importance measure for the component.

The Birnbaums structural importance measure for component A in a 'don't care phase' fault tree is calculated by:

$$I(A) = \sum_{i=1}^{m_A} I(A^i) \cdot w_{A^i}, \tag{4.2.11}$$

where m_A is the number of failure modes in which component A can fail in and A^i is the variable representing the failure of component A in failure mode i and its importance measure, $I(A^i)$, is calculated by using Equation (4.2.12), where m_{max} is the maximal number of failure modes experienced by any component in the 'don't care phase' fault tree.

$$I(A^i) = Q(1_{A^i}, \mathbf{q}) - Q(0_{A^i}, \mathbf{q}). \tag{4.2.12}$$

- $Q(1_{A^i}, \mathbf{q})$ is the top event probability with probability 1 for event A^i, probability 0 for any event A^j $(i \neq j)$ and probability $q = \frac{1}{m_{max}+1}$ for any of the remaining events.
- $Q(0_{A^i}, \mathbf{q})$ is the top event probability with probability 0 for event A^i and probability $q = \frac{1}{m_{max}+1}$ for any of the remaining events.

The weight of A^i is calculated to be:

$$w_{A^i} = \frac{1}{m_A}. \tag{4.2.13}$$

A variable(component) is ordered earlier if it:

1. Has the highest Birnbaums structural importance measure value. In case of ties;
2. Appears earlier in the "don' t' care phase' fault tree in a top-down, left-right manner.

The 'don't' care phase' fault tree of the mission failure fault tree illustrated in Figure 4.2.2 is shown in Figure 4.2.5. The maximal number of failure modes experienced by any component in the fault tree in Figure 4.2.5 is 3. Therefore, all variables are assigned with probability 0.25 to calculate the value of $Q(1_{A^i}, \mathbf{q})$ and $Q(0_{A^i}, \mathbf{q})$. The importance

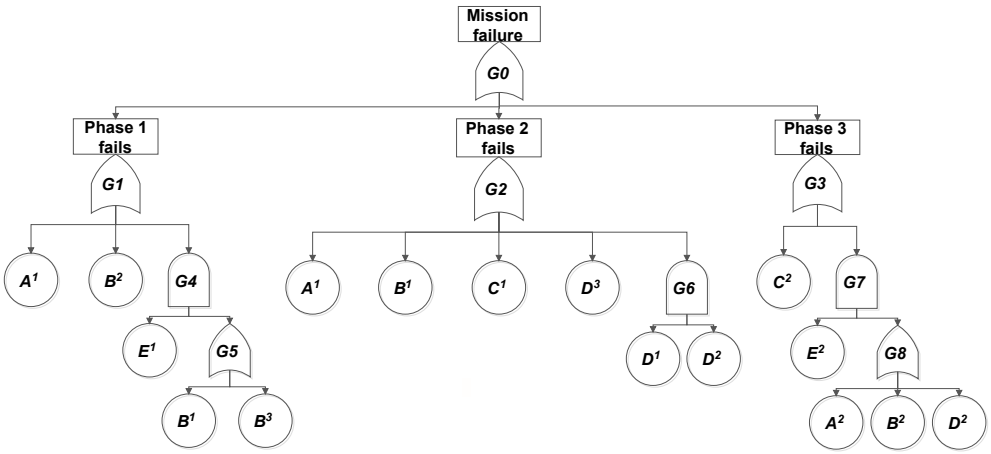

Figure 4.2.5: The 'don't care phase' fault tree for mission failure fault tree in Figure 4.2.2

measure for each component is calculated as:

$$
\begin{aligned}
I(A) &= \frac{I(A^1) + I(A^2)}{2} = 0.0742 \\
I(B) &= \frac{I(B^1) + I(B^2) + I(B^3)}{3} = 0.807 \\
I(C) &= \frac{I(C^1) + I(C^2)}{2} = 0.103 \\
I(D) &= \frac{I(D^1) + I(D^2) + I(D^3)}{3} = 0.0104 \\
I(E) &= \frac{I(E^1) + I(E^2)}{2} = 0.0546.
\end{aligned}
\tag{4.2.14}
$$

The variables are ordered by decreasing value of I, which gives the order list:

$$B < C < A < E < D. \tag{4.2.15}$$

4.3 The Proposed Best Order Interleaving (BOI) Scheme

4.3.1 Motivation for the Development of BOI

This section outlines an ordering scheme that has been developed during the course of this research to provide an ordering scheme that is suitable for use in a reliability analysis process that can be used to support decision making in autonomous systems.

A PMS might not complete its original phases due to the unacceptable system reliability resulting from changes of system conditions. In this case, other mission configurations must be considered in order to identify an acceptable alternative. The tasks involved in performing future missions may differ from those of the original mission, meaning that

the construction of the BDDs required to calculate the unreliability of future phases and of the mission as a whole might require a new variable ordering scheme to be developed in order to ensure efficiency in the calculations required.

All of the previous 8 schemes initially require the mission configuration being defined to form the mission failure fault tree before sorting variables according to specified rules. When a mission alternative needs to be considered, variables must be re-ordered and previous phases up to the current point of the mission must be re-analysed before continuing to calculate the reliability of the following phases. Due to the fact that new components may be included because of the involvement of changing system functionality, continuing to use the original order list could lead to analysis failure or a dramatic increase in BDD sizes, and hence a large increase in computation time. Therefore, it is necessary to re-order variables in the alternative mission configuration. However, the variable re-ordering and reconstruction of the BDDs of previously-performed phases will slow down the reliability analysis of mission alternatives and delay the decision making process of identifying an optimal mission alternative.

A Online-offline strategy that was suggested in order to speed up the decision making process involved carrying out as much of the construction and computation process as possible offline, before the mission begin, to reduce the amount of computation required online, once the mission is underway [31]. This would require variable ordering schemes to be derived and BDDs to be constructed before the mission begins. However, when considering decisions that must be made in response to events that occur during the course of a mission, the requirement to obtain the mission failure fault tree in order to apply the eight ordering schemes presented in Section 4.2 has made this strategy impossible (as the mission failure fault tree of the mission alternative is only obtained once the mission is underway). In order to be able to evaluate the options available and make decisions as to the best next course of action as quickly as possible by employing the online strategy, a new ordering scheme, BOI, is proposed.

BOI arranges variables according to the structures of the fault trees representing the failure of tasks that PMS is capable of performing rather than the structure of the mission failure fault tree. Therefore, variables can be ordered before the mission starts and the order list can remain unchanged no matter how the mission configuration changes since the variable ordering principles of BOI ensure all variables are arranged in such a way that all possible mission alternatives can be considered. The principles of the BOI

scheme allow a strategy to be adopted where the maximum possible amount of time to computation is carried out offline. This will save a large amount of time to re-order variables and reconstruct BDDs when performing reliability analysis for an alternative mission configuration once the mission is underway, when compared with the analysis using the previously-investigated ordering schemes. BDDs representing the failure of the PMS to perform specific tasks can be constructed in advance, meaning that decisions on following actions can be made more rapidly. Supposing the PMS has successfully completed m phases of the original mission, then the source of the advantage of the BOI scheme is illustrated in Figure 4.3.1.

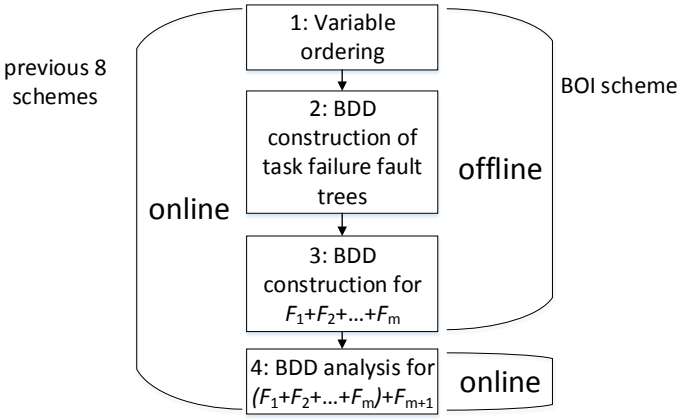

Figure 4.3.1: Illustration of the source of the efficiency advantage of BOI over the other schemes

4.3.2 The Description of the BOI

The proposed BOI scheme uses an interleaving technique, which was introduced to provide an optimal order list for multiple output combinational circuits [39]. The technique is used in the BOI scheme to combine order lists that lead to the smallest BDD size for each task failure fault tree to obtain an overall optimal order list for all possible variables. The obtained order list will remain unchanged no matter how the mission configuration changes after the mission has started, as all possible variables are guaranteed to be arranged positions in the list.

The ordering schemes described in Section 4.2 can be classified into two groups according to the characteristics of the rules followed when applying them:

Group 1: Scheme 1, Scheme 5, Scheme 6, Scheme 8.

Group 2: Scheme 2, Scheme 3, Scheme 4, Scheme 7.

In group 1 schemes, variables are ordered in a global range according to an assigned value of certain parameters. In group 2 schemes, variables are explored in a depth-first manner, i.e., variables below a gate are fully allocated before the exploration of another gate. This common principle of group 2 schemes provides a basis for applying the interleaving technique. Therefore, the BOI schemes use group 2 schemes to obtain optimal variable order lists for individual task failure fault trees in order that they can then be integrated using the interleaving technique.

The steps involved in ordering variables when using the BOI scheme are shown in Figure 4.3.2. The steps of the proposed BOI ordering scheme are described below:

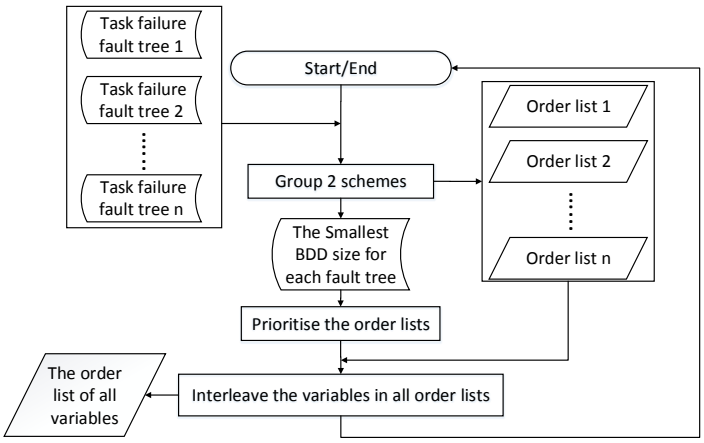

Figure 4.3.2: The steps of BOI ordering

1. Read in all fault trees representing the failure of the tasks of which the system is capable.

2. Use the group 2 schemes to analyse each of the fault trees, thus obtaining 4 BDDs for each fault tree.

3. Compare the sizes of the 4 BDDs and select the optimal ordering scheme, i.e., that which leads to the smallest BDD sizes, for each fault tree and cache the optimal order list and the smallest BDD size for each fault tree.

4. Prioritise the order lists according to decreasing size of the smallest BDD corresponding to each fault trees.

5. Interleave variables in the order lists using the following interleaving technique to form a final list, according to the following process until all variables are included in

the final list (for the first order list considered, variables are simply copied into the final list):

(a) For the first variable in the order list, if the variable is already in the final list, then do nothing; otherwise, insert it at the beginning of the final list.

(b) For the other variables in the order list, check whether the variable, (assume without loss of generality) A, is in the final list, if so, do nothing; otherwise, identify the position in the final list of the variable B that is immediately in front of variable A in the current order list and insert variable A immediately after variable B in the final list.

4.3.3 Example

Consider the UAV described previously, which is capable of performing 4 different tasks whose failure logic is presented by the fault trees in Figure 4.2.1, and is required to perform mission M_A, whose failure logic is shown by the fault tree in Figure 4.2.2. When using BOI to order variables, the fault tree structures representing the failure of the system to complete all tasks of which it is capable is used, rather than the fault tree structure of the phased mission the system is required to perform. Applying the group 2 schemes to each of the task fault trees in Figure 4.2.1, the optimal order list that gives the smallest BDD size for each task failure fault tree can be obtained. The optimal order list and the smallest BDD size for each task failure fault tree are shown in Table 4.3.1.

Task failure fault tree	Optimal order list	Smallest BDD size
T1	$B < A < E$	6
T2	$D < A < B < C$	4
T3	$C < E < A < D < B$	5
T4	$F < E < A < D$	6

Table 4.3.1: The optimal order list and the smallest BDD size obtained by applying group 2 schemes to each of the task failure fault trees shown in Figure 4.2.1

The order lists for T1 and T4 lead to the same smallest sizes of BDDs, in this case, the order lists are given the priority as the tasks labelled, i.e. T1 is considered before T4, denoted as T1<T4. The priority of order lists are: T1 < T4 < T3 < T2. Therefore, first of all, all components in the order list for T1 are added to the final list. The final list then contains three components: $B < A < E$. Then the order list of T4 is integrated into the final list. Since the first component F in the order list of T4 is not yet in final list, it is allocated to the first position in the final list. Components E and A are already in final

list, meaning that no action need to be taken. Since the last component D in the order list for T4 appears immediately after A in that order list, D is inserted into the final list immediately after A. Therefore, the interleaving of the order list for T4 with the order list for T1 gives the updated final list: $F < B < A < D < E$. Next, the order list of T3 is integrated into the updated final list. C is the first component in the order list for T3 and is not yet in final list; therefore, it is allocated to the first position of the final list. This means all components have now been allocated a position in the final list and there is no need to consider the remaining variables and order lists. The BOI scheme therefore gives the following order list:

$$C < F < B < A < D < E. \tag{4.3.1}$$

To illustrate the potential advantage to be gained by using the BOI scheme, consider the following situation. Suppose during the performance of phase 3 of mission M_A, an internal failure is detected which leads to the abortion of M_A and consideration of an alternative mission, M_B, with task 1 as phase 1, task 2 as phase 2, task 3 as phase 3 and task 4 as phase 4. Scheme 1 to Scheme 8 sorted variables according to the fault tree structure of M_A as shown in Figure 4.2.2 and the variable order lists provided by Scheme 1 to Scheme 8 will no longer be applicable because of the involvement of a new component F due to the inclusion of the new phase 4. Therefore, if any of Scheme 1 to Scheme 8 is to be adopted during the process of quantification of phase and mission unreliability for mission M_B, all variables must be re-ordered and BDDs up to phase 3 reconstructed according to the new variable ordering before calculating the unreliability of phase 4 and the unreliability of M_B as a whole. The variable ordering and BDD construction process takes time that might otherwise be used for quantification and hence may impact the speed of the decision making. However, BOI develops a variable ordering scheme that applies to all the task failure fault tree structures shown in Figure 4.2.1 and since it does not depend on a specific mission failure fault tree structure, it can be applied to any mission configuration. This means that no matter how the mission configuration changes, the order list produced using BOI remains unchanged and therefore, no time is spent re-ordering variables and reconstructing BDDs representing $F_1 + F_2 + F_3$ can be saved and more rapid decision making process is allowed.

4.4 Comparison and Conclusion

The ordering schemes are tested on phased missions generated using the same method described in Section 3.4. A comparison of analysis efficiency is made by measuring the sizes of the BDDs constructed and the time taken to perform quantitative analysis in order to find the conditional unreliability of phases and the mission unreliability. Firstly, the analysis efficiency of the nine ordering schemes described in Section. 4.2 and Section. 4.3 will be compared. Then, the advantage of the proposed BOI scheme will be demonstrated. The section ends by drawing conclusions about the results obtained.

Since Model 3 proved to be the most efficient BDD model for the tested phased missions, Model 3 is used to construct and quantify the BDDs for individual phase failures and mission failure. Using the nine ordering schemes and Model 3, nine BDDs are obtained for each of the phased missions. The sizes of the BDDs representing mission failure for each tested phased mission, constructed by using each of the ordering scheme, are shown in Table B.2.1. The times using each ordering scheme to analyse each mission contains three parts: the time used for variable ordering and converting task fault trees to BDDs, the calculation the conditional unreliability of each phase and the calculation of the unreliability of the whole mission. The analysis times including all these three parts using the nine ordering scheme for all tested missions are shown in Table B.2.2. The time is counted using the clock_t function in C++ coding language. Section 4.4.1 contains a comparison of the nine ordering schemes under consideration, which was the data in Table B.2.1 and Table B.2.2.

4.4.1 Comparison of the Nine Ordering Schemes

In Table 4.4.1, the second row shows the number of missions for which each scheme produced the smallest BDD; the third row shows the number of missions for which each scheme resulted in the lowest mission analysis time. The results in Table 4.4.1 illustrate that it is possible for any scheme to produce the smallest BDD. When considering analysis time, Scheme 6 and Scheme 8 never result in the lowest time, since the variable ordering rules required when applying Scheme 6 and Scheme 8 consume much more time compared with the other ordering schemes. This is because Scheme 6 requires Scheme 5 to be applied to a series of dynamically changing fault trees until all components are ordered and Scheme 8 needs repeated calculation of top event probabilities in order to obtain the Birnbaum' s importance measure for each basic event in the mission fault tree. Of the nine schemes,

Scheme 5 shows increased performance efficiency in comparison to the others in terms of both BDD size and analysis time for mission failures. As illustrated in Table 4.4.1, Scheme 5 results in the smallest mission failure BDD in 30 out of the 100 missions studied and the lowest analysis time in 48 out of the 100 missions studied. This is a higher chance of access than any of the other ordering schemes.

	Scheme 1	Scheme 2	Scheme 3	Scheme 4	Scheme 5	Scheme 6	Scheme 7	Scheme 8	BOI
BDD size	4	4	10	4	30	10	19	16	3
Analysis time	8	11	22	6	48	0	4	0	1

Table 4.4.1: Performance comparison of the nine ordering schemes measured by size of BDD and analysis time for mission unreliability

Since the time taken to quantify mission and phase unreliability is a key factor in using reliability analysis technique to support decision making, the performance efficiency of the ordering schemes is compared only using analysis time in the remainder of this section.

The efficiency advantages in terms of analysis time of each scheme over the others are given in Table 4.4.2 to Table 4.4.10. The tables demonstrate how much faster the overall analysis is when analysing each ordering scheme, i.e, the average percentage of time saved by using one scheme in preference to the other schemes for the 100 tested phased missions. For example, Table 4.4.2 shows the average percentage of time saved analysing mission unreliability using Scheme 1 as opposed to the other schemes. It can be clearly seen that scheme 1 is less efficient than Scheme 2, Scheme 3, and Scheme 5 while it is more efficient than the others. Table 4.4.3 shows the average percentage of time saved analysing mission unreliability using Scheme 2 as opposed to the others. It can be seen that Scheme 2 is less efficient than Schemes 1,3,4,5 and 6 but more efficient, on average, than the others.

	Scheme 2	Scheme 3	Scheme 4	Scheme 5	Scheme 6	Scheme 7	Scheme 8	BOI
Mean	-15.38%	-41.25%	2.54%	-70.72%	17.36%	13.00%	45.37%	27.82%

Table 4.4.2: Efficiency advantage of Scheme 1 over the other schemes in terms of time taken to analyse mission failure

	Scheme 1	Scheme 3	Scheme 4	Scheme 5	Scheme 6	Scheme 7	Scheme 8	BOI
Mean	-52.86%	-34.88%	-25.25%	-144.76%	-20.33%	15.42%	16.69%	36.28%

Table 4.4.3: Efficiency advantage of Scheme 2 over the other schemes in terms of time taken to analyse mission failure

As shown in Table 4.4.6, the average percentage of time saved analysing mission unreliability using Scheme 5 as opposed to the others is positive in each case. This indicates that Scheme 5 can be relied upon to result in less mission unreliability analysis time than

	Scheme 1	Scheme 2	Scheme 4	Scheme 5	Scheme 6	Scheme 7	Scheme 8	BOI
Mean	-16.33%	13.84%	14.85%	-78.96%	16.16%	31.14%	46.23%	44.08%

Table 4.4.4: Efficiency advantage of Scheme 3 over the other schemes in terms of time taken to analyse mission failure

	Scheme 1	Scheme 2	Scheme 3	Scheme 5	Scheme 6	Scheme 7	Scheme 8	BOI
Mean	-46.46%	-38.13%	-63.24%	-115.07%	0.23%	4.43%	38.85%	16.08%

Table 4.4.5: Efficiency advantage of Scheme 4 over the other schemes in terms of time taken to analyse mission failure

all the other schemes on average.

	Scheme 1	Scheme 2	Scheme 3	Scheme 4	Scheme 6	Scheme 7	Scheme 8	BOI
Mean	24.20%	18.60%	7.68%	34.89%	49.25%	40.15%	67.56%	44.60%

Table 4.4.6: Efficiency advantage of Scheme 5 over the other schemes in terms of time taken to analyse mission failure

The proposed BOI schemes can be seen to perform very poorly in terms of the time taken to perform mission unreliability analysis, as shown in Table 4.4.10. The average percentages of reduced time for analysing mission unreliability using BIO are all negative numbers, which indicates, using BIO to analyse the mission failure probabilities will spend more time compared with the analysis time using the other schemes.

4.4.2 Advantage of the Proposed BOI Scheme

The proposed BOI scheme was shown to perform inefficiently in comparison to the other ordering schemes when analysing the unreliability of a mission, as illustrated in Table 4.4.10. However, the BOI scheme has been developed with a view to enabling the efficient use of the reliability analysis of PMS in a decision making tool for autonomous systems. It is expected to improve analysis efficiency when calculating the failure probability of possible alternative missions compared to the mission currently being performed, in the event that the probability of failure of the current mission drops to an unacceptable level and alternatives must be considered. In particularly, this will be the case for the first altered phase in the alternative mission is required to perform a new task that is not included in current mission.

In this case, the unreliability analysis of the first altered phase which is a task that contains new components that the current mission does not, (meaning that variable re-ordering is compulsory and the previous phases up to the current point of the mission must be re-analysed) requires the alternative mission configuration being defined to form

	Scheme 1	Scheme 2	Scheme 3	Scheme 4	Scheme 5	Scheme 7	Scheme 8	BOI
Mean	-72.89%	-73.45%	-102.34%	-36.35%	-132.11%	-23.33%	34.71%	-20.66%

Table 4.4.7: Efficiency advantage of Scheme 6 over the other schemes in terms of time taken to analyse mission failure

	Scheme 1	Scheme 2	Scheme 3	Scheme 4	Scheme 5	Scheme 6	Scheme 8	BOI
Mean	-97.54%	-60.49%	-126.96%	-56.38%	-229.25%	-47.96%	-8.96%	1.51%

Table 4.4.8: Efficiency advantage of Scheme 7 over the other schemes in terms of time taken to analyse mission failure

the mission failure fault tree before sorting variables when using Scheme 1 to Scheme 8. This means that when calculating the analysis time required following the definition of an alternative mission configuration, it is necessary to consider the time needed for variable ordering, converting task fault trees to BDDs and calculation of the unreliability of completed phases shared by the current and alternative missions up to the current point. For the analysis using BOI, these tasks do not contribute to the online analysis time, since the variable ordering and task fault tree conversion have been performed offline and the calculation of the unreliability of completed phases has already been performed while analysing the original mission. Therefore, only the time taken to analyse the first altered phase is counted when using the BOI scheme.

In reality, phased mission systems such as UAVs may have higher chances of failure during the earlier or later phases of a mission. For example, due to the higher probability of failure during take off or landing, two cases are considered here to investigate the possible advantage that can be gained from using the BOI scheme to order variables prior to the system analysis.

Case 1: Alternative mission configurations must be considered at the end of the second phase. The time taken to analyse the unreliability of the first altered phase in the mission alternative, which in this case is the third phase, using the nine ordering schemes is shown in Table B.2.3.

Case 2: Alternative mission configurations must be considered at the start of the penultimate phase. The time taken to analyse the unreliability of the first altered phase

	Scheme 1	Scheme 2	Scheme 3	Scheme 4	Scheme 5	Scheme 6	Scheme 7	BOI
Mean	-270.61%	-262.87%	-310.83%	-163.57%	-364.99%	-100.89%	-142.40%	-151.14%

Table 4.4.9: Efficiency advantage of Scheme 8 over the other schemes in terms of time taken to analyse mission failure

	Scheme 1	Scheme 2	Scheme 3	Scheme 4	Scheme 5	Scheme 6	Scheme 7	Scheme 8
Mean	-236.74%	-131.98%	-290.76%	-192.54%	-547.40%	-197.88%	-78.56%	-131.02%

Table 4.4.10: Efficiency advantage of BOI over the other schemes in terms of time taken to analyse mission failure

in the mission alternative, which in this case is the penultimate phase, using the nine ordering schemes is shown in Table B.2.4.

The results presented in Table B.2.3 and Table B.2.4 demonstrate the advantage that can be gained by using the BOI scheme. In most cases, the analysis time when using BOI is much less than that when using the other schemes. This is because the BOI scheme avoids the need to reorder variables and construct BDDs with a new variable ordering for those mission phases shared by the original and alternative mission.

The average efficiency advantages (percentage reduction in analysis time) of using BOI in preference to the other ordering schemes to analyse the unreliability of the first altered phase, in the two cases outlined above are shown in Table 4.4.11. The first row gives the average efficiency advantage of the BOI scheme when analysing the unreliability of the third phase in the mission alternative, and the second row gives the average efficiency advantage of the BOI scheme when analysing the unreliability of the penultimate phase of the mission alternative. The average efficiency advantage of the BOI scheme over the other schemes varies from 39.33% to 95.13%. In all cases, BOI leads to a lower unreliability analysis time for the first altered phase in the alternative mission.

How much more efficient BOI is compared to the other schemes, depends on the time needed when applying the other schemes to reorder variables and then recalculating probabilities that have already been computed (which will increase the relative efficiency of BOI). It will also depend on the time needed to quantitatively analyse the failure probability of the first altered phase (which may reduce the relative efficiency of BOI; as shown in Table 4.4.10, BOI usually leads to slower reliability analysis for mission failures compared with other ordering schemes). Therefore, the efficiency advantage of BOI is not guaranteed to increase when the mission configuration changes in a later phase, because although the time saved for reordering variables and recalculating probabilities of previous phases using BOI is more when the mission configuration changes in a later phase compared the time saved when mission configuration changes in an earlier phase, the time needed to analyse the first altered phases using BOI in this case may also be longer. If the time needed to analyse the first altered phase using BOI is much longer than when using the

other schemes, the advantage of the BOI scheme over the other schemes may be offset. For example, consider the efficiency advantage of the BOI scheme compared to Scheme 4; the average percentage of reduction in time decreased from 69.95% to 56.62%.

	Scheme 1	Scheme 2	Scheme 3	Scheme 4	Scheme 5	Scheme 6	Scheme 7	Scheme 8
The third phase	47.96%	40.19%	39.33%	69.95%	48.40%	83.21%	76.44%	95.13%
The penultimate phase	45.42%	49.13%	44.04%	56.62%	43.47%	63.21%	64.23%	75.62%

Table 4.4.11: The average efficiency advantages of using BOI to analyse the first altered phase (the third/penultimate phase)of the mission alternative comparing with using the other ordering schemes

When an urgent decision is needed as to the best next course of action so that the PMS with a serious system failure can respond immediately to the condition changes in order to avoid platform damage or crash, the computation of the unreliability of the first altered phase of the mission alternative must be performed in the shortest possible time. The less time that is needed, the faster a decision can be made and the faster the system can respond. The results presented in this section suggest that BOI has great potential for reducing analysis time when considering alternative mission configurations and hence could improve the speed of updated reliability analysis when used as a part of a decision making tool.

4.4.3 Summary

The construction of BDDs initially requires variables to be ordered and how they are ordered can greatly affect the sizes of the constructed BDDs. The research in this chapter has extended eight schemes that were previously applied to standard fault trees with a single phase and single failure mode components to the construction of BDDs for PMS with multiple failure mode components. A new ordering scheme, BOI, is proposed, which is designed to work efficiently within the decision making tool described in Section 2.6. It is specifically developed to enable the updated reliability analysis, which is performed when alternative mission configurations must be considered while the mission is underway, to be performed more quickly.

After testing the ordering schemes on 100 sample missions, Scheme 5 was seen to offer greater average efficiency than the other schemes in terms of both BDD size and analysis time, as shown in Table 4.4.1 and Table 4.4.6.

Although the BOI scheme was shown to be relatively insufficient when compared to the other schemes when considering only BDD size and time taken to perform quantitative

analysis of the mission, Table 4.4.11 shows that the BOI scheme is, on average, more efficient than the other schemes when considering the calculation of the unreliability of the first altered phase in a new mission in the context of updated unreliability calculation within a decision making tool of the type presented in Section 2.6. Two cases were used to demonstrate this advantage, considering phases towards the start and the end of the mission being modelled. In both cases, BOI was seen to offer increases in efficiency, with up to 95% reduction in analysis time compared to the other techniques considered.

The proposed BOI scheme therefore demonstrates a great efficiency advantage in terms of analysis time for calculating the reliability of the first altered phase in an alternative mission configuration. This comes about because the variable order list produced by applying the BOI scheme remains the same no matter how the mission configuration changes and no time must therefore be spent reordering variables or reapplying successfully completed phases in contrast to the other ordering schemes.

When a mission is underway, the PMS might not be able to complete its original mission phases due to changes in the system or environmental conditions, in which cases, other mission configurations must be considered in order to identify an acceptable alternative. When there is sufficient time to identify an appropriate mission alternative (to allow analysis of the entire mission unreliability rather than analysis of the first altered phase in an alternative mission), Scheme 5 appears to be the best choice of ordering scheme to use in the analysis, as it has been proved to be the one with highest chance of producing the smallest BDD sizes for mission failure and mission unreliability analysis time. When time is limited, i.e, the configuration of the next phase needs to be decided almost immediately, and particularly in the case when variable reordering would be necessary if other ordering schemes were used (for example, new components are involved in the task to be performed), the BOI scheme is recommended to be used since it avoids the need for variable reordering and re-analysis of phases shared by the mission underway and the mission alternative under consideration, so that the unreliability of the first altered phase of the mission alternative can be quickly computed and an acceptable mission configuration can be decided in the shortest possible time.

Chapter 5

Development of BDD-Based Approximation Methods for PMS with Multiple Failure Mode Components

5.1 Introduction

A reliability analysis method that allows fast, accurate phased mission reliability quantification could be used to quickly analyse a number of alternative mission scenarios, in addition to an originally planned mission, and therefore allow a comparison of the relative chances of success and an informed decision to be made as to the best next course of action for a system following a failure event. The speed of quantification would be particularly important for systems where decision making must be performed quickly, such as a UAV. Consider for example, an engine fire has been detected when a UAV is performing a reconnaissance mission, in which case, continuing its current mission would lead to an uncontrolled crash, an immediate decision as to the best next course of action is required to ensure the UAV can land safely.

In cases such as these, the time available to perform updated unreliability analysis is limited, and even using the quantification techniques and ordering schemes described in the previous chapters, the time taken to analyse BDDs to find the desired unreliabilities may be too great to be of practical use. It is may therefore be necessary to be able to make a trade off between the accuracy and speed of the calculation performed. It is feasible that in

some situations, solutions may be obtained to an accuracy that is sufficient to form a basis for decision making by using approximation. Such approximation may also be obtained more quickly than exact solutions and are hence have great potential to be used as part of the updated reliability analysis within a decision making process. Approximations should therefore be investigated to find the impact of their use in obtaining updated reliabilities in as little time as possible with reasonable accuracy loss.

Approximations of the reliability of PMS were firstly introduced in [47], where lower and upper bounds for mission unreliability were developed and the conservative approximations (upper bounds of mission unreliability) are compared in [12]. However, because of the dependencies that arises due to component failure in different phases in the PMS, the approximations can be very inaccurate.

Thus, models that can provide fast, accurate reliability approximations for PMS containing components with multiple failure modes must be developed if they are to support the real-time decision making process described in Section 2.6. Since the efficiency of BDD analysis for PMS has already been demonstrated in the previous chapters, BDD models may also offer the greatest potential for developing a method that allows reasonably accurate, approximated results to be obtained more quickly than the exact results.

In this chapter, three approximation methods using BDDs have been proposed:

1. Approximation by multiplication of individual phase unreliability, after a review of the upper bounds of PMS introduced in [47].

2. A rare event approximation model that involves constructing and quantifying zero-suppressed BDDs (Z-BDDs), BDDs that encode only minimal cut sets [20], for mission failure fault trees, after a review of the Z-BDD model for analysis of standard fault trees with single phase and single failure mode components [17].

3. Approximation model using truncated BDDs, defined as BDDs with nodes replaced by terminal 0 nodes at some point during construction when the maximum of probabilities of all paths from the root node down to the current node at which the truncation taken place are smaller than a truncation limit, after a review of a truncated BDD model for analysis of standard fault trees [18].

The approximation models are developed to be used in the online-offline strategy to form a methodology that could support real-time decision making in the most efficient and accurate way as possible. After the three approximation models have been developed,

their efficiency and accuracy are compared by testing on a number of PMS and conclusions are drawn as to which of these models are the most promising for use in a decision-making process.

5.2 Approximation Using the Early Stage Method

The early stage approximation methods demonstrated in [46] [47] [12] are for analysis of non-repairable PMSs and the mission reliability is estimated by the product of the reliability of all mission phases. Several upper bound approximations for mission unreliability are presented. In this section, all these upper bounds are first reviewed and then BDDs are introduced to decrease the computational effort for mission unreliabilities. Following this the online-offline strategy is concluded to develop an approximation methodology for use in updated reliability analysis as part of a decision making process.

5.2.1 Literature Review of Early Stage Approximation Method

In [14], the success of the coherent phased mission system (the minimal cut sets of fault trees representing coherent systems contain only failures of basic events), is represented by the intersection of the success of each phase and several bounds for the mission reliability are studied, since they require less computational effect than quantification of the exact results. In [12], rather than reliability bounds, the corresponding unreliability upper bounds of non-repairable PMSs are computed using fault trees.

5.2.1.1 Approximation Bounds for Mission Failure Probability

In [12], the researchers compared the accuracy of the mission unreliability upper bounds developed in [14]. The four upper bounds for mission unreliability are as follows:

5.2.1.2 INEX

Using traditional fault tree analysis, the unreliability of each phase, $P(F_i)$, is calculated by identifying the identified minimal cut sets and applying the inclusion-exclusion expansion (Equation (2.2.21)) with the probability of basic events calculated using the failure probability density function of the components. Suppose NMC_i is the number of minimal cut sets for F_i, C_j^i is the j^{th} minimal cut set of F_i, the unreliability of F_i, $P(F_i)$ is calculated

using Equation (5.2.1).

$$P(F_i) = P(\sum_{j=1}^{NMC_i} C_j^i) \tag{5.2.1}$$

Suppose there are n phases in total in the mission. The INEX upper bound, denoted as P_{INEX}^{UPPER} is calculated by multiplication of the probabilities of success of all n phases, and then subtracted from 1 to obtain the upper bound for the mission unreliability, as shown by Equation (5.2.2):

$$P_{INEX}^{UPPER} = 1 - \Pi_{i=1}^{n}(1 - P(F_i)) \tag{5.2.2}$$

5.2.1.3 INEX-CC

The INEX-CC upper bound is obtained in a similar way to the INEX upper bound, except before the computation of $P(F_i)$ for each phase i, minimal cut cancellation [46] is performed to ease the computational burden associated with calculating the terms of the inclusive-exclusive expansion. A minimal cut set C_1 of a previous phase can be cancelled if each element in C_2, which is a minimal cut set of a later phase, is also in C_1, i.e., C_1 is an subset of C_2 ($C_1 \subset C_2$). The INDEX-CC upper bound is denoted by $P_{INEX-CC}^{UPPER}$.

5.2.1.4 Minimal Cut Bound (MCB) Method

The MCB method gives the minimal cut set upper bound for mission unreliability. It is obtained by substituting component failure probabilities to Equation (5.2.3) to obtain an approximation for the unreliability of each phase.

$$P(F_i) = \Pi_{j=1}^{NMC_i}(1 - P(C_j^i)) \tag{5.2.3}$$

where NMC_i is the number of minimal cut sets for F_i. The expression for $P(F_i)$ is then substituted into Equation (5.2.2) to give the MCB upper bound for the mission unreliability.

5.2.1.5 Minimal Cuts Bound with Cut Cancellation(MCB-CC)

The MCB-CC upper bound is obtained using the same process as is used to calculate the MCB except before the phase unreliability computation, cut set cancellation is performed to reduce the number of minimal cut sets and therefore reduce the computational effort

for each $P(F_i)$.

5.2.1.6 Comparing the Bounds

the following relationships hold between the four approximations to the mission unreliability described above and the exact mission unreliability, Q_{miss} [47][12]:

$$Q_{miss} \leq P_{INEX-CC}^{UPPER} \leq \begin{matrix} P_{MCB-CC} \\ P_{INEX}^{UPPER} \end{matrix} \leq P_{MCB}. \tag{5.2.4}$$

No general comparison can be made between P_{MCB-CC} and P_{INEX}^{UPPER}, since the order of these two values depends on the problem being solved [12].

5.2.2 Using BDDs to Implement Early Stage Approximation (Method 1)

The BDD models described in Section 2.3 provide exact analysis results while needing less computational effort compared with the traditional fault tree analysis approach of applying the inclusion-exclusion expansion to the acquired minimal cut sets, for systems with single phase and single failure mode components represented by fault trees. Since in analysis of PMS, the unreliability of entire mission can be approximated using the unreliability of each phase without considering the dependencies across different phases (as presented in Section 5.2.1), BDD analysis could be used to replace the traditional inclusion-exclusion approach to expedite the computation of individual phase reliability. By converting task failure fault trees of the PMS into BDDs, the unreliability of each mission phase i, $P(F_i)$ can be easily calculated BDD analysis. Then, the INEX upper bound of mission unreliability, P_{INEX}^{UPPER}, can be obtained by substituting the phase unreliability into Equation (5.2.2).

The steps of using method 1 to approximate the entire mission unreliability in the context of the online-offline strategy are as follows:

1. (Offline) Convert all task failure fault trees into BDDs using Equation (2.3.3) and store them. No global variable ordering scheme is required, since the dependencies across phases are not considered and the variable ordering scheme that leads to the smallest BDD size is preferred for each task failure fault tree, so that the quantification process can be performed as fast as possible in the online stage.

2. (Online) Calculate the unreliability of each phase, $P(F_i)$, using Equation (2.3.19) and updated variable probabilities.

3. (Online) Substitute the acquired $P(F_i)$ into Equation (5.2.2) to obtain the INEX upper bound for the mission unreliability, $P^{UPPER}_{INEX-CC}$.

5.2.3 Summary

Method 1 approximates the mission unreliability by computing the unreliability (reliability) for individual mission phases separately rather than analysing one single equivalent fault tree structure representing the entire mission failure. Generally, this will require less computational effort, since the the BDD structure for each phase is already constructed offline and the quantification of BDDs can be quickly carried out online once a mission is defined.

Method 1 has a great efficiency advantage when execution decisions are urgently needed, since the BDDs for the task failure fault trees can be constructed and stored offline with only BDD quantification required online, which could be performed very quickly. However, decisions made based on the analysis results of Method 1 should be used with care, as the dependencies that arise between variables relating to the same components but across different phases can lead to significant accuracy loss [46] and thus potencially inaccurate and unsound decisions.

5.3 Rare Event Approximation

The rare event approximation is accurate if the probabilities of variables in minimal cut sets are rare, since in this case the contribution of the later terms in Equation (2.2.24) is relatively small when compared with the contribution of the first term, so that they can be ignored without big impact on the final approximation value.

Due to the efficiency advantage that BDD models bring to fault tree analysis, BDD models might be expected to show promise in computing approximations to mission reliability that could be used to support urgent decision making for PMS when a failure has been detected and updated probabilities should be calculated.

In [33], the researchers developed an algorithm to obtain the Z-BDDs, BDD structures that encode only the minimal cut sets, by applying a $WITHOUT$ operator to the exact BDD obtained by direct conversion from a fault tree, which was detailed in Section 2.3.3. The researchers in [17] developed a novel algorithm, which constructs the Z-BDDs for coherent fault trees by directly applying the $WITHOUT$ operator to BDDs under construction. The Z-BDD, which is usually smaller than the normal BDD (since the Z-

BDD only encodes MCS), is constructed directly using the developed algorithm and this is expected to lead to increased efficiency in terms of both memory storage and time for coherent system.

In this section, a novel algorithm is developed to convert the coherent fault trees representing the failures of PMS containing components with multiple failure mode to Z-BDDs, based on the algorithm for converting standard fault trees to Z-BDDs presented in [17]. A quantification method that can then be used to calculate the rare event approximation to the probability of mission failure by quantifying the Z-BDD is proposed together with the Z-BDD algorithm.

This section starts with a review of the Z-BDD algorithm for standard fault trees, and then a novel Z-BDD model, which is used to compute the mission unreliability for PMS with multiple failure mode components is presented in the context of providing a reliability methodology that could be used in a real-time decision support tool for PMS underway.

5.3.1 Literature Review of Z-BDD Algorithm for Standard Fault Trees

In [17], a Z-BDD algorithm, which directly convert coherent fault trees where no NOT gates are involved, to Z-BDDs. It works by trunacting the BDD as necessary during the construction. The obtained Z-BDDs encode only the minimal cut sets.

5.3.1.1 Z-BDD Construction Rules

For coherent systems, the Shannon's decomposition as defined in Equation (2.3.2) could be simplified as follows[17]:

$$f = ite < x, f_1, f_0 >= xf_1 + f_0. \tag{5.3.1}$$

Consider two BDD nodes: $F = ite < x, F1, F0 >$ and $G = ite < y, G1, G0 >$, where $x \leq y$. The Z-BDD construction rules for the Boolean operation of the two BDDs nodes of a coherent fault tree are:

$$F \cdot G = \begin{cases} ite < x, F1 \cdot G1 + F1 \cdot G0 + F0 \cdot G1, F0 \cdot G0 > & if\ x = y \\ ite < x, F1 \cdot G, F0 \cdot G > & if\ x < y \end{cases} \tag{5.3.2}$$

$$F+G=\begin{cases} ite<x,F1+G1,F0+G0> & if\ x=y \\ ite<x,F1,F0+G> & if\ x<y \end{cases} \tag{5.3.3}$$

The subsuming calculation, i.e. $WITHOUT$ operator, described in Section 2.3.3, can then be applied to the intermediate BDDs under construction when using this algorithm so that faster computation of minima cut sets is achieved and less memory space is consumed.

5.3.1.2 Example

Consider the fault tree in Figure 5.3.1, the exact BDD constructed using the usual rules given in Equation (2.3.3) is shown in Figure 5.3.2. By applying the $WITHOUT$ operator described in Section 2.3.3 to the exact BDD in Figure 5.3.2, the Z-BDD shown in Figure 5.3.3 is obtained. If applying the Z-BDD algorithm described above directly to the fault tree in Figure 5.3.1, using the same variable order list as the exact BDD does: $A<B<C$, the Z-BDD for the fault tree in Figure 5.3.1 can be constructed as follows:

$$\begin{aligned} G2 &= A\cdot B = ite<A,1,0>\cdot ite<B,1,0>= ite<A,ite<B,1,0>,0> \\ G1 &= C+G2 = ite<C,1,0>+ite<A,ite<B,1,0>,0> \\ &= ite<A,ite<B,1,0>,ite<C,1,0>> \end{aligned} \tag{5.3.4}$$

It can be seen that the Z-BDD obtained using the Z-BDD algorithm is the same as the Z-BDD shown in Figure 5.3.3, which was obtained by applying the $WITHOUT$ operator to the exact BDD. Both algorithms produce Z-BDDs that can be used to obtain the minimal cut sets: $\{A,B\}$ and $\{C\}$.

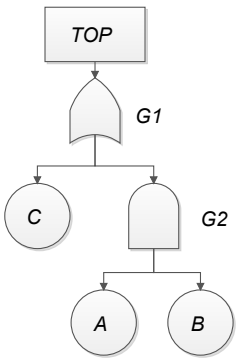

Figure 5.3.1: An example fault tree

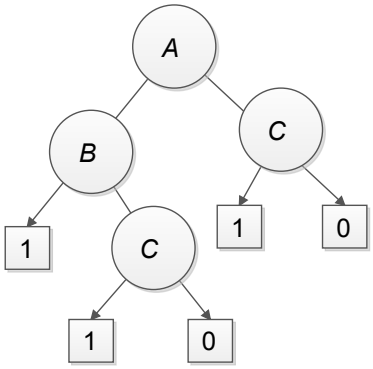

Figure 5.3.2: The exact BDD

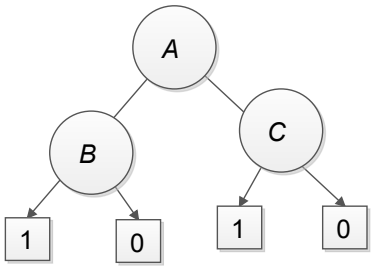

Figure 5.3.3: The Z-BDD after truncation

5.3.1.3 Summary

The Z-BDD algorithm developed in [33] initially requires conversion from fault tree to BDD before application of the $WITHOUT$ operator to this exact BDDs. Considering the purpose of this research presented in this thesis, which is to develop a method to find the mission reliability in the shortest possible time in order to support decision making, the less time needed for analysis using the approximation model, the more desirable the model is. By applying the developed algorithm to fault trees, the Z-BDDs containing minimal cut sets are achieved directly, which can save computational effort and time when compared with the first approach. In order to take advantage of the ability to obtain Z-BDDs directly from fault trees, an algorithm must be developed that can convert fault trees representing the failure of PMS containing multiple failure mode components to Z-BDDs and achieve approximations to the mission unreliability directly from the acquired Z-BDD.

5.3.2 Rare Event Approximation using Z-BDDs for PMS (Method 2)

In this section, a novel algorithm has been developed to construct the Z-BDD for a coherent phased mission fault tree and then perform quantification to obtain the rare event approximation of the mission unreliability.

5.3.2.1 Development of Z-BDD Construction Rules for PMS

For coherent fault trees representing the overall mission failure of a PMS, $(F_1+F_2+\cdots F_n)$, the Z-BDDs encoding the minimal cut sets can be derived from the Shannon decomposition presented in Equation (5.3.1).

The algorithm for Z-BDD construction from fault trees representing PMS is given

below: suppose $F = ite < x, F1, F0 >$, $G = ite < y, G1, G0 >$ and $x \leq y$,

$$F{\cdot}G = \begin{cases} ite < x, F1 \cdot G1 + F1 \cdot G0 + F0 \cdot G1, F0 \cdot G0 > & x = y \\ ite < y, F1 \cdot G1 + F0 \cdot G1, G0 \cdot F > & cp(x) = cp(y), fm(x) = fm(y), pn(x) \neq pn(y) \\ ite < x, F1 \cdot G0, F0 \cdot G > & cp(x) = cp(y), fm(x) \neq fm(y) \\ ite < x, F1 \cdot G, F0 \cdot G > & cp(x) \neq cp(y) \end{cases} \tag{5.3.5}$$

$$F + G = \begin{cases} ite < x, F1 + G1, F0 + G0 > & x = y \\ ite < x, F1, F0 + G > & \text{Else} \end{cases}$$

The proof of cases of $x = y$ and $cp(x) \neq cp(y)$ are given in [17]. The proofs of the other cases are given below.

Proof 5.3.1 *When $cp(x) = cp(y)$, $fm(x) = fm(y)$, and $pn(x) > pn(y)$ since $x \leq y$*

$$\begin{aligned} & ite < x, F1, F0 > \cdot ite < y, G1, G0 > \\ =& (x \cdot F1 + F0) \cdot (y \cdot G1 + G0) \\ =& x \cdot y \cdot F1 \cdot G1 + x \cdot F1 \cdot G0 + y \cdot G1 \cdot F0 + F0 \cdot G0 \\ & (x \cdot y = y \text{ by Equation } (3.1.1)) \\ =& y \cdot F1 \cdot G1 + x \cdot F1 \cdot G0 + y \cdot F0 \cdot G1 + F0 \cdot G0 \\ =& x \cdot F1 \cdot G0 + y \cdot F1 \cdot G1 + F0 \cdot (y \cdot G1 + G0) \\ =& x \cdot F1 \cdot G0 + y \cdot F1 \cdot G1 + F0 \cdot G \\ =& ite < x, F1 \cdot G0, ite < y, F1 \cdot G1, F0 \cdot G >> . \end{aligned} \tag{5.3.6}$$

$$\begin{aligned} & ite < x, F1, F0 > + ite < y, G1, G0 > \\ =& x \cdot F1 + F0 + G \\ =& ite < x, F1, F0 + G > \end{aligned} \tag{5.3.7}$$

When $cp(x) = cp(y)$ *and* $fm(x) \neq fm(y)$,

$$
\begin{aligned}
& ite < x, F1, F0 > \cdot ite < y, G1, G0 > \\
=& (x \cdot F1 + F0) \cdot (y \cdot G1 + G0) \\
=& x \cdot y \cdot F1 \cdot G1 + x \cdot F1 \cdot G0 + y \cdot G1 \cdot F0 + F0 \cdot G0 \\
& (x \cdot y = 0 \ \textit{by Equation (3.1.2)}) \\
=& x \cdot F1 \cdot G0 + F0 \cdot (y \cdot G1 + G0) \\
=& x \cdot F1 \cdot G0 + F0 \cdot G \\
=& ite < x, F1 \cdot G0, F0 \cdot G >
\end{aligned}
\tag{5.3.8}
$$

$$
\begin{aligned}
& ite < x, F1, F0 > + ite < y, G1, G0 > \\
=& x \cdot F1 + F0 + G \\
=& ite < x, F1, F0 + G >
\end{aligned}
\tag{5.3.9}
$$

The $WITHOUT$ operator can then be applied during the construction of the Z-BDD so that an smaller BDD can be always obtained and therefore Z-BDD can be obtained in less time and with less memory space consumption.

The error between the exact result and rare event approximation is negligible if failure events are rare. For updated phase or mission reliability analysis, if the failure of a certain component or subsystem is identified, i.e, the failure probability of the component or subsystem is 1, which means the event is no longer rare, to ensure analysis accuracy, a reduction process needs to be performed to the obtained Z-BDD where all nodes whose variables with high failure probability (close to 1) are replaced by their 1-branch and all nodes with variables that relate to the same component as the variables identified to have high failure probabilities but relating to different failure modes are replaced by their 0-branch.

Consider the Z-BDD shown in Figure 5.3.4 and assume that component B, which has two failure modes, 1 and 2, has been identified to fail in failure mode 1. Applying the reduction process, node $N1$ with variable B_1 is replaced by its 1-branch, and the BDD resulting from this step is shown to the middle of the figure. Then, B_2 is replaced by its 0-branch, and the BDD following this reduction is shown to the right of the figure.

By eliminating all variables with probability close to 1 and those relating to the same

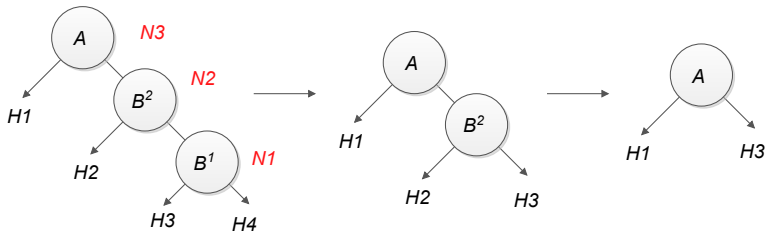

Figure 5.3.4: Failure event removal

component but in a different failure mode, the Z-BDD representing mission failure can then be quantified to obtain an accurate rare event approximation to the mission unreliability.

5.3.2.2 Quantification of Z-BDDs

The rare event approximation, which is an upper bound for the top event probability of a fault tree representing system failure, is equal to the sum of the probabilities of all minimal cut sets for the fault tree, as presented in Equation (2.2.24). The rare event approximation of a coherent mission failure fault tree can therefore be calculated by quantifying the Z-BDD described above, ignoring variables representing the success of events.

Consider the BDD node illustrated in Figure 3.2.2 from a Z-BDD. The 1-branch always links two variables that relate to the same failure mode and component but different phase number, or that relate to different components. The probability of node F can therefore be calculated using Equation (5.3.10):

$$P(F) = \begin{cases} p(x)P(F1) + P(F0) & cp(x) \neq cp(y) \\ p(y)P(H1) + p(x)P(H0) + P(G0) & cp(x) = cp(y), fm(x) = fm(y) \end{cases} \tag{5.3.10}$$

Proof 5.3.2 *When $cp(x) \neq cp(y)$, the situation is the same as that considered in [33] and when $cp(x) = cp(y)$ and $fm(x) = fm(y)$, $P(G) = E[x \cdot G1] + E[G0]$ and*

$$\begin{aligned} E[x \cdot G1] &= E[x \cdot (y \cdot H1 + H0)] \\ &E[x \cdot y \cdot H1 + x \cdot H0] \\ &= E[y \cdot H1 + x \cdot H0] \\ &= p(y)P(H1) + p(x)P(H0_{x=1}) \\ P(G) &= p(y) \cdot P(H1) + p(x) \cdot P(H0_{x=1}) + P(G0) \end{aligned} \tag{5.3.11}$$

where $H0_{x=1}$ is the first node encountered during the traversal down the success branch of $G1$ with variable which belongs to a different components with x. Substituting Equation (5.3.11) *into $P(G)$ to obtain the case when $cp(x) = cp(y)$ and $fm(x) = fm(y)$ of Equation* (5.3.10).

5.3.3 Summary

The developed Z-BDD construction and quantification method makes it possible to directly convert fault trees representing the overall mission failure of PMS containing multiple failure mode components to Z-BDDs and the rare event approximation of the mission failure probability can be quickly obtained by quantifying the root node of the Z-BDDs using Equation (5.3.10). The constructed Z-BDDs are expected to be smaller than the exact BDDs and the direct quantification of a Z-BDD to achieve the rare event approximation of the top event probability for the mission failure fault tree could lead to a much faster reliability analysis than when computing the exact value.

Method 2 is hence expected to provide an upper bound estimation (rare event approximation) to the mission unreliability in a shorter time compared with exact quantification. Thus, Method 2 might be used as a reliability approximation model to provide mission unreliability approximations for many mission alternatives in as short a time as possible and thus help to make a decision as to the best next course of action for a system undergoing a phased mission to choose the optimal alternative.

However, the approximation method is based on the rare event approximation. Therefore, when probabilities of the variables are not rare, Method 2 should be used with care. It may, thus, be necessary to investigate other approximation methods.

5.4 Approximation Using Truncated BDDs

In [18], an algorithm to reduce the size of the BDD during conversion from a fault tree without sacrificing the solution probability accuracy was developed, since the top event probability rapidly converges to an exact value according to a lowered truncation limit. The smaller size of BDD obtained requires less quantification time to be spent on and the method also allows the error of the approximation to be controlled in a relatively small range.

5.4.1 Literature Review of BDD Truncation for Standard Fault Trees

Truncation of a BDD is performed by considering a truncation limit (TL) into the BDD construction process. A parameter called upper probability, p, is incorporated into the BDD construction rules, whose value is compared with the value of TL whenever a BDD operation is performed in order to determine a truncation point.

The rules for constructing truncated BDDs for standard fault trees are described as follows: consider two nodes: $F = ite < x, F1, F0 >$, and $G = ite < y, G1, G0 >$, where $x \leq y$. Then the operation of the two nodes, $F \diamond G$ with respect to the current value of upper probability p is:

$$(F \diamond G, p) = \begin{cases} 0 & \text{if } p < TL \\ ite < x, (F1 \diamond G1, p * p_x), (F0 \diamond G0, p * q_x) > & \text{if } p \geq TL, \text{ and } x = y \\ ite < x, (F1 \diamond G, p * p_x), (F0 \diamond G, p * q_x) >, & \text{if } p \leq TL, \text{ and } x < y \end{cases} \tag{5.4.1}$$

The BDD construction and upper probability calculation are performed simultaneously in a top-down manner. When the first BDD operation starts, the upper probability is given value 1, $p \leftarrow 1$; each time a 1-branch operation, $F1 \diamond G1$ or $F1 \diamond G$, is performed, the upper probability is multiplied by the occurrence probability of variable x (corresponding to the probability of the basic event occurrence in the considered fault tree), $p \leftarrow p * p_x$; each time a 0-branch, $F0 \diamond G0$ or $F0 \diamond G$, is performed, the upper probability is multiplied by the non-occurrence probability of variable x (corresponding to the probability of the basic event non-occurrence in the considered fault tree), $p \leftarrow p * q_x$, where $q_x = 1 - p_x$. i.e., The upper probability for standard fault trees is calculated as follows:

1. $p \leftarrow 1$ when BDD operation starts.
2. $p \leftarrow p * p_x$ when the 1-branch operation, $F1 \diamond G1$ or $F1 \diamond G$, is performed.
3. $p \leftarrow p * (1 - p_x)$ when the 0-branch operation, $F0 \diamond G0$ or $F0 \diamond G$, is performed.

Whenever the upper probability, p, of operation $F \diamond G$ is less than the pre-defined truncation limit, TL, the operation will return terminal 0 node, meaning that any following BDD operations will not continue, and therefore leading to a smaller BDD being obtained compared with the exact BDD constructed using the same rules.

The upper probability is so-called because, in the computation table, the operation of $F \diamond G$, always stores the largest p, as $F \diamond G$ can be calculated for more than one time and therefore several p values can appear. In the algorithm, each operation $F \diamond G$, its upper probability value p and the resulting BDD node, H, is identified by F, G and the Boolean operator $\diamond$ in the computation table. Consider the computation of $F \diamond G$ with upper probability p:

$F \diamond G$ is already computed and stored in the computation table: If the upper probability of the stored operation $F \diamond G$, p' is bigger than p, then, the stored result BDD node $R' = F \diamond G$ is returned; Else (if the stored p' is smaller than p), $F \diamond G$ is recomputed and the operation result in the computation table is updated by replacing the existing $R' = F \diamond G$, p' with the new computed BDD node, R, and the bigger upper probability p, i.e., $R = F \diamond G$, p.

$F \diamond G$ is not yet computed: Insert $F \diamond G$ into the computation table, together with the computed BDD node R and the upper probability p .

The technique of always storing the larger upper probability ensures that the stored BDD node result R is always bigger than the BDD to be calculated for the same operation $F \diamond G$ and therefore, the computation table can be kept at a small size.

Also, by using the bigger result BDD node for the operation $F \diamond G$, meaning that less truncation is performed, accurate solution could possibly be obtained.

5.4.2 Summary

The development of the truncated BDD algorithm makes it possible to build a small BDD for a standard fault tree and the benchmark tests in [18] demonstrated the efficiency of the developed method. The approximation of the top event probability obtained through quantifying the truncated BDD converges to the exact value rapidly as the truncated limit is decreased.

Although the use of the truncation limit can effectively reduce the size of BDDs, the repeated calculation of $F \diamond G$ because of the different values of the upper probability p can lead to long-winded analysis time [25]. For example, suppose operation $(F \diamond G, 0.1)$ takes t seconds and during the lower level operation, no node is replaced by 0. When $(F \diamond G, 0.11)$ is computed, the whole operation process will be repeated without any change of BDD nodes to ensure that $F \diamond G$ generate a bigger result BDD node. Also, the effect of the truncation

limit on the top event probability is unknown, since by replacing certain BDD nodes with terminal 0 nodes, some paths that used to end at terminal 1 nodes will now end at terminal node 0, meaning that those paths that are supposed to contribute to the top event failure probability will now be neglected. The number of eliminated paths is unknown and thus the impact of ignoring those paths on the top event probability is unknown. Last but not least, unreasonable selection of truncation limit (where it is set at too low a value) may lead the sizes of truncated BDDs to be even bigger than the exact BDDs, leading to higher memory consumption and longer analysis time, which makes the approximation process meaningless. This is due to the fact that the replacement of terminal 0 nodes may lead to new result BDD nodes and thus offset the efficiency advantage of the sub-node sharing feature of BDD analysis[25].

5.5 Truncated BDD Method for PMS (Method 3)

With the aim of developing an approximation method that can be used to support rapid decision making for a PMS operating in a changing environment, a new truncated BDD method is developed, which constructs the truncated BDDs for PMS containing multiple failure mode components.

A truncated BDD algorithm, which can evaluate the top event probability of a standard fault tree more efficiently than the exact BDD analysis with reasonable losses in accuracy, is reviewed in Section 5.4.1. However, no research has been carried out to investigate approximations for phased mission systems using the BDD truncation method. In order to develop a truncated BDD method for phased missions, the component dependencies that arise due to phase operation and multiple failure modes must be considered. In addition to the development of a novel BDD truncation model for PMS, an analytical technique is developed to give a truncation limit value, which can be used to control the error between the approximated value and exact value of the root node probability (to a required precision).

In this section, the computation of the analytical truncation limits for PMSs depending on the error between the exact and approximated values of the top event probability of the the considered fault tree and the structure of the considered fault tree is investigated. Then, a novel algorithm, which is used to construct truncated BDDs for PMS is developed. Finally, a technique is introduced to calculate lower and upper bounds for the conditional unreliability of mission phases and the unreliability of the entire mission.

5.5.1 The Assignment of a Truncation Limit

Suppose N_i is the i^{th} BDD node that is replaced by a terminal 0 node, and $P_1, P_2, \cdots, P_{n_i}$ are the n_i paths leading to node N_i from the root node, as illustrated in Figure 5.5.1. The

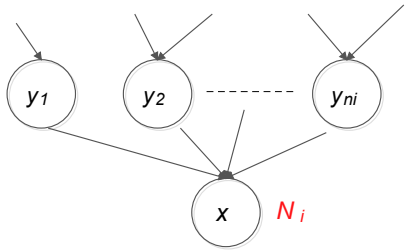

Figure 5.5.1: An example truncated BDD node N_i

error, defined as the difference between the exact and approximated values of the root node probability caused by replacing node N_i by 0, is denoted by δ_i. If each parent node of N_i has a variable that relate to a component other than N_i, meaning that each variable at the end of path P_j, $j = 1, 2, \cdots, n_i$, is independent of the variable of N_i ($cp(x) \neq cp(y_i)$, for $i = 1, 2, \cdots n_i$ in Figure 5.5.1), δ_i can be represented as:

$$\delta_i = [P(P_1) + P(P_2) + \cdots + P(P_{n_i})] * P(N_i), \tag{5.5.1}$$

suppose there are m nodes in total that are replaced by terminal 0 nodes under truncation limit TL, then the error between the exact and approximated values is given as follows (the probability of the root node could be calculated by the sum of the probabilities of all paths ending at a terminal 1 node [31]):

$$\delta = \sum_{i=1}^{m} \delta_i \tag{5.5.2}$$

It is possible to define a TL, which can control the error between the approximated and exact values of the root node probability of the considered truncated BDD to a specific level of precision, 10^{-q}, where $q \geq 2$. Substituting Equation (5.5.1) to Equation (5.5.2)

gives:

$$\delta = \sum_{i=1}^{m}[P(N_i) * \sum_{k=1}^{n_i} P(P_k)]$$

$$(max(P(P_1), P(P_2) \cdots, P(P_{n_i}) < TL) \Rightarrow \delta < TL * \sum_{i=1}^{m} P(N_i) * n_i \tag{5.5.3}$$

$$(0 \leq P(N_i) \leq 1) \Rightarrow \delta < TL * \sum_{i=1}^{m} n_i$$

Since n_i is the number of paths that go into node N_i, and N_i could either be a terminal 1 node or a non-terminal node, the total number of paths into N_i, $\sum_{i=1}^{m} n_i$, is less than the total number of paths ending at terminal 1 nodes in the BDD representing a failure occurrence between the start of the mission and the end of phase p ($F_1 + F_2 + \cdots + F_p$). Denoting NP_j^1 as the total number of paths ending at terminal 1 nodes in the BDD representing failure of mission phase j, F_j, (equivalent to the total number of paths in the BDD equivalent to the fault tree representing the failure of the corresponding task) and NP_j^0 as the total number of paths ending at terminal 0 nodes in the same fault tree representing F_j, then:

$$\sum_{i=1}^{m} n_i < \begin{cases} NP_1^1 & \text{For the BDD representing } F_1 \\ NP_1^1 + \sum_{i=2}^{p} NP_i^1 \prod_{j=1}^{i-1} NP_j^0 & \text{For the BDD representing } F_1 + F_2 + \cdots F_p \end{cases} \tag{5.5.4}$$

Now, denote the error between the exact and approximated values for the root node probability is ϵ, in order to control the error, i.e., $\delta < \epsilon$, the TL must satisfy:

$$TL * \sum_{i=1}^{m} n_i < \epsilon. \tag{5.5.5}$$

Therefore,

$$TL < \frac{\epsilon}{NP_1^1 \text{ or } NP_1^1 + \sum_{i=2}^{p} NP_i^1 \prod_{j=1}^{i-1} NP_j^0}. \tag{5.5.6}$$

Consider for example if the unreliability of a mission is of the order 10^{-2}, and the magnitude of the approximation error is required to be 10^{-3}, i.e., $\epsilon = 10^{-3}$. Using Equation

(5.5.6), the analytical truncation limit for different phases is

$$TL < \frac{10^{-3}}{NP_1^1 \text{ or } NP_1^1 + \sum_{i=2}^{p} NP_i^1 \prod_{j=1}^{i-1} NP_j^0}. \tag{5.5.7}$$

It can be seen that the truncation limit will decrease as the mission continues and more phases are involved, due to the greater number of paths involved.

5.5.1.1 The Application of the Analytical Truncation Limit Value

The OR gate ratio can be quite high in fault trees of task failures relating to the real-world systems [29]. This implies that after the task failure fault trees are converted to BDDs, there should be more terminal 1 nodes than terminal 0 nodes and thus for expression $NP_1^1 + \sum_{i=2}^{p} NP_i^1 \prod_{j=1}^{i-1} NP_j^0$, a linearly-increasing rate is expected rather than an exponentially-increasing rate when considering the total number of paths ending at terminal 1 nodes. The analytical truncation limits are then expected to be of a reasonable magnitude.

Adopting the analytical TL allows the error between the approximate value obtained using the truncated BDD analysis and the exact value of the top event probability to be controlled. When the order of magnitude of the analytical TL is reasonably small, the size of the constructed BDD can be smaller than the exact BDD and therefore more efficient reliability analysis can be achieved.

The error analysis is only correct when the variable of the node to be truncated relates to a different component than the variable of the adjacent node above, since otherwise, Equation (5.5.1) would not be true due to the dependencies between variables relate to the same component but different phase indices and failure modes.

The developed analytical truncation limit is applicable solely to control the error between the lower bound approximation using the following described algorithm and the exact entire mission unreliability, since the assumptions used in the derivation of the analytical TL only applies for terminal 0 node replacement.

5.5.2 Development of Truncated BDD Algorithm

In fault trees representing PMS containing multiple failure mode components, dependencies arise due to different failure modes and the contribution of events to failure in different mission phases, which makes the truncation BDD method reviewed in Section 5.4.1 for standard fault trees unsuitable to use. Therefore, a novel algorithm is developed in this

section to construct truncated BDDs while considering these dependencies so that the quantification methods described in Chapter 3 can be used to quantify the truncated BDDs.

The main idea of the truncation is that when constructing a BDD, if the upper probability of an operation of two BDDs nodes is smaller than a pre-set truncation limit TL, then the operation returns a terminal 0 node. An upper probability p is assigned for every operation of two BDD nodes. In BDDs for PMS, dependencies arise between variables that relate to the same component failure in different phases or failure modes, meaning that the upper probability which quantifies the contribution of the current path to the root node probability can no longer be calculated by simple multiplication of variables tracing from the root node down to the current operation.

5.5.2.1 The Truncated BDD Algorithm Accounting for Dependency

To address the dependencies existing between variables and thus perform an appropriate upper probability calculation for an operation of two BDD nodes, a Boolean expression γ is introduced. γ could either be a boolean variable, or an AND operation of two or more variables that relate to the same component on the path from the root node down to the current operation, calculated using Equation (3.1.1) and Equation (3.1.2).

γ is used to solve dependencies existing between variables relating to the same component so that a simplified form of the operation of the variables can be obtained in order to calculate an appropriate upper bound. It is also used to decide whether the component of γ, which is defined as the component to which the variables in γ relate, is different from the component to which the next variable to be considered relates. If the two components are different, meaning that the two Boolean expressions are independent of each other, then the probability of γ is multiplied by the current upper probability to obtain an updated upper probability.

The introduction of γ ensures that the upper probability is calculated while considering dependencies between variables so that sound decisions can be made as to whether to truncate at the computed BDD node. The currently-dependent variables are stored in γ temporarily with dependencies addressed, the component of γ is then compared with that of the next variable to be considered in order to decide whether to update the current upper probability by multiplying by the probability of the current γ or to include the new variable in the current γ and use Equation (3.1.1) and Equation (3.1.2) to solve the

dependencies that arise.

The probability of γ is calculated according to the appropriate basic event failure probability density functions and Equation (3.1.4). The Boolean variable γ and the upper probability, p form a new data structure $\theta =< p, \gamma >$, which is needed in order to calculate an appropriate upper probability, i.e., p is the upper probability used to determine whether the operation to be performed will return a result of terminal 0 node, while γ is used to solve dependencies between variables on the path from the root node down to current operation to ensure the accuracy of the calculation of p. The value of θ is known whenever an operation of two BDD nodes is performed.

Instead of using the product of the upper probability and the probability of the current γ, $p * P(\gamma)$, which is the probability of the path tracing from the root node down to the current operation, as the parameter value to determine whether truncation should occur, using the upper probability p ensures that the BDD node to be truncated has a variable that relates to a different component than the last variable on the path from the root node down to the current BDD operation. This means that for the same PMS fault tree considered, a smaller size of truncated BDD will always lead to a smaller computed root node probability (approximation to the top event probability), since smaller BDD means more paths are reduced and thus the error between the approximate and the exact values increases according to Equation (5.5.1).

Consider an operation of two BDD nodes ($F \diamond G, \theta =< p, \gamma >$), where $F = ite < x, F1, F0 >$ and $G = ite < y, G1, G0 >$, under a pre-set truncation limit TL, and $p \leftarrow 1$, $\gamma \leftarrow 1$ when the BDD operation starts. The proposed truncated-BDD algorithm is illustrated in Algorithm. 5.1.

5.5.2.2 The Calculation of θ_u and θ_v

The θ_u and θ_v for 1-branch and 0-branch operations in step 15 and step 16 in Algorithm 5.1 are calculated according to Algorithm 5.2 and Algorithm 5.3, by considering the dependency between variable x and the Boolean expression γ, which can be solved using Equation (**??**) and Equation (3.1.2). For the variable x, let $cp(x)$, $fm(x)$, $pn_{start}(x)$ and $pn_{end}(x)$ be the component and the failure mode to which x relates to, the starting and the ending phase index during which period x occurs.

For the DEP-BDD model, the 1-branch always link two variables that relate to two different components, or the same component and failure mode, the 0-branch always link

Algorithm 5.1 truncated-bdd

Input: $\diamond$, F, G, $\theta =< p, \gamma >$.

{/*Terminal cases*/}
if $p < TL$ **then** return 0.
else if $(F = 0)$ or $(G = 0)$ or $(F = 1)$ or $(G = 1)$ **then**
  $R \leftarrow$ truth-table($\diamond$, F, G)
end if
{/*Compute table already contains $F \diamond G$*/}
if computation-table-have($(\diamond, F, G) = H, \theta' =< p', \gamma' >$) and $p < p'$ **then**
  $R \leftarrow H$.
end if
if $x > y$ **then** swap(F,G)
end if
{/*Compute ($F \diamond G$, θ), where $F =< x, F1, F0 >$, $G =< y, G1, G0 >$ */}
{/*Compute 1-branch u and 0-branch v, and their θ_u and θ_v*/}
$\theta_u \leftarrow compute\theta_u(x, \theta)$
$\theta_v \leftarrow compute\theta_v(x, \theta)$
if x=y **then**
  $u \leftarrow$ truncated-bdd($< \diamond, F1, G1 >, \theta_u$)
  $v \leftarrow$ truncated-bdd ($< \diamond, F0, G0 >, \theta_v$)
else if $cp(x) \neq cp(y)$ **then**
  $u \leftarrow$ truncated-bdd($< \diamond, F1, G >, \theta_u$)
  $v \leftarrow$ truncated-bdd ($< \diamond, F0, G >, \theta_v$)
else if $cp(x) = cp(y)$, $fm(x) \neq fm(y)$ **then**
  $u \leftarrow$ truncated-bdd($< \diamond, F1, L1 >, \theta_u$)
  $v \leftarrow$ truncated-bdd ($< \diamond, F0, G >, \theta_v$)
else
  $u \leftarrow$ truncated-bdd($< \diamond, F1, G >, \theta_u$)
  $v \leftarrow$ truncated-bdd ($< \diamond, F0, G0 >, \theta_v$)
end if
{/*Search or add the obtained BDD node H into ite-table*/}
$R \leftarrow$Search-add-ite-table(x, u, v).
{/*Search or add the obtained BDD node H into compute-table*/}
Update-insert-compute-table ($(\diamond, F, G)$=R, $\theta =< p, \gamma >$).

Return: R.

to two variables that relate to two different components, or the same component but different failure modes [43]. For the BDD model in [36], the 1-branch always link two variables that relate to two different components, the 0-branch always link to two variables that relate to two different components, or the same component and failure mode, or the same component but different failure mode.

Therefore in a BDD constructed using the truncation algorithm, the 1-branch always link two variables x and y (assume without loss of generality $x < y$) that satisfy $[cp(x) \neq cp(y)]$, or $[cp(x) = cp(y),\ fm(x) = fm(y)$, and $pn_{end}(x) > pn_{end}(y)]$, the 0-branch always link to two variables satisfy $[cp(x) \neq cp(y)]$, or $[cp(x) = cp(y),\ fm(x) \neq fm(y)]$, or $[cp(x) = cp(y),\ fm(x) = fm(y),\ pn_{end}(x) < pn_{end}(y)]$. This relationships are used to compute θ_u and θ_v.

The operation on a path involves only the AND operation. θ_u and θ_v are calculated according to the dependencies existing between γ and the variable to be involved, x.

Consider first θ_u for the 1-branch operation. When $\gamma = 1$, p remains and $\gamma_u \leftarrow x$. When $cp(x) = cp(\gamma)$, if γ is a single variable (in this case, γ satisfies $fm(x) = fm(\gamma)$ and relates to its occurrence state), then according to the first formula of Equation (**??**), $\gamma \cdot x = x$; else, i.e., $fm(x) \neq fm(\gamma)$ (in this case, the variables in γ relates to their non-occurrence states), according to the second equation in Equation (3.1.2), $\gamma \cdot x = x$. Thus, $\gamma_u \leftarrow x$ and the value of p remains. Consider for instance: $A^1_{02} \cdot A^1_{01} = A^1_{01}$ and $A^{\bar{3}}_{02} \cdot A^{\bar{2}}_{02} \cdot A^1_{01} = A^1_{01}$. When $cp(x) \neq cp(\gamma)$, $p_u \leftarrow p * P(\gamma)$, because the two adjoint variables relate to two different components and are independent of each other, and $\gamma_u \leftarrow x$. The algorithm of θ_u computation is given in Algorithm. 5.2.

Algorithm 5.2 compute θ_u

Input: x, $\theta =< p, \gamma >$.
1: **if** $\gamma = 1$ or $cp(x) = cp(\gamma)$ **then**
2: $\{p$ remains; x is integrated in γ to obtain the $\gamma_u \leftarrow \gamma \cdot x\}$
3: $p_u \leftarrow p$.
4: $\gamma_u \leftarrow x$.
5: **else**$\{cp(x) \neq cp(\gamma)\}$
6: $p_u \leftarrow p * P(\gamma)$.
7: $\gamma_u \leftarrow x$.
8: **end if**
Return: $\theta_u =< p_u, \gamma_u >$

Now consider θ_v for the 0-branch operation. When $\gamma = 1$, p remains and $\gamma \leftarrow \bar{x}$. When $cp(x) = cp(\gamma)$, p remains, suppose z is the last variable in γ, if z relate to its non-occurrence state, then if $fm(x) = fm(z)$, then according to the second formula of the phase algebra,

Equation (**??**), $z \cdot \bar{x} = \bar{x}$ and thus $z \leftarrow \bar{x}$ and all the other variables in γ remains if any. For example, consider $\overline{A^2_{01}} \cdot \overline{A^1_{01}} \cdot \overline{A^1_{02}} = \overline{A^2_{01}} \cdot \overline{A^1_{02}}$ (this case can only exist in Forward-BDD model); else if $fm(x) \neq fm(z)$, then according to the fourth formula in dependency algebra, Equation (3.1.2), $\gamma_v \leftarrow \gamma \cdot \bar{x}$, for example, consider $(\bar{A^3_{02}} \cdot \bar{A^2_{02}}) \cdot \bar{A^1_{01}} = \bar{A^3_{02}} \cdot \bar{A^2_{02}} \cdot \bar{A^1_{01}}$. Else, i.e., z relates to its occurrence state (also implies z is the only variable in γ), then if $fm(x) = fm(z)$, then according to the third formula in phase algebra, $\gamma_v \leftarrow x'$, where x' satisfies $x' \leftarrow z$, $pn_{start}(x') \leftarrow pn_{end}(x)$, $pn_{end}(x') \leftarrow pn_{end}(z)$, for example, $A^2_{03} \cdot \bar{A^2_{02}} = A^2_{23}$ (this case can only exist in DEP-BDD model); else, i.e., $fm(x) \neq fm(z)$, according to the second equation in dependency algebra, $\gamma_v \leftarrow \gamma \cdot \bar{x} \leftarrow \gamma$, for example, consider $A^1_{23} \cdot \overline{A^2_{02}} = A^1_{23}$. When $cp(x) \neq cp(\gamma)$, the situation is the same with θ_u.

Algorithm 5.3 compute θ_v

Input: $\bar{x}$, $\theta =< p, \gamma >$

if $\gamma = 1$ or $cp(x) = cp(\gamma)$ **then**
{/*p remains; x, is integrated in γ to obtain the $\theta_u.\gamma \leftarrow \theta.\gamma \cdot x$*/}
  $p_v \leftarrow p$.
  if $\gamma = 1$ **then** $\gamma_v \leftarrow \bar{x}$
  end if
  z is the last Boolean variable in γ
  if z relates to its non-occurrence state **then**
    if $fm(x) = fm(z)$ **then** $\gamma_v \leftarrow \gamma$, $z \leftarrow \bar{x}$
    else $\gamma_v \leftarrow \gamma \cdot \bar{x}$
    end if
  else{/*z relate to its occurrence state */}
    if $fm(x) = fm(z)$ **then**
      $x' \leftarrow z$
      $pn_{start}(x') \leftarrow pn_{end}(x)$
      $\gamma_v \leftarrow x'$
    else $\gamma_v \leftarrow z$
    end if
  end if
else{/*$cp(x) \neq cp(\gamma)$*/}
  $p_v \leftarrow p * P(\gamma)$
  $\gamma_v \leftarrow \bar{x}$.
end if

Return: $\theta_v =< p_v, \gamma_v >$

5.5.3 Quantification of Truncated Phase BDDs

The truncated BDDs developed using the algorithm in the above section and any of the BDD construction models presented in Chapter 3 can be quantitatively analysed using the quantification method corresponding to the BDD model used, i.e., if the truncated BDDs

are built using BDD Model 1 or BDD Model 2, the truncated BDDs can be quantified using the DEP-BDD evaluation described in Section 3.2.2; if the truncated BDDs are built according to the construction rules of BDD Model 3, then they can be evaluated by the quantification method described in Section 3.3.3.

5.5.3.1 Lower and Upper Bound Approximations

The truncated BDDs constructed using the developed truncation algorithm replace some of the terminal 1 nodes and non-terminal nodes in the exact BDDs by terminal 0 nodes to obtain smaller BDDs. Since the probabilities of the replaced BDD nodes are always positive, the approximation obtained when quantifying the truncated BDD thus provides a lower bound for the exact probability of the top event representing an entire mission failure or the occurrence of a failure between the start of the mission and the end of current phase.

Approximation Method 3 could also be used to produce an upper bound for the exact probability of the top event occurrence. In [23], researchers demonstrated the use of another truncated BDD, which would give an upper bound approximation to the top event probability by replacing all of the nodes whose upper probability is smaller than a pre-defined truncation limit with a terminal 1 node instead of a terminal 0 node.

The algorithm developed in Section 5.5.2.1 can be used to construct truncated BDDs that provide an upper bound or a lower bound approximation to the exact probability of the top event that representing a failure for a PMS containing multiple failure modes, the only difference would be when an truncation point is reached, the operation of two BDD nodes return a terminal 0 node or a terminal 1 node.

The truncated BDDs with truncated nodes replaced by terminal 0 nodes are denoted as Zero-oriented-truncated BDDs (ZT-BDDs) and by quantifying which will provide a lower bound for the top event occurrence probability. The truncated BDDs with truncated nodes replaced by terminal 1 nodes are denoted as One-oriented-truncated BDDs (OT-BDDs) and by quantifying which will provide an upper bound for the top event occurrence probability.

5.5.4 Approximations to Conditional Unreliabilities of Mission Phases

Both of the lower and upper bound approximations to mission unreliability, or the probability of the event representing a system failure has occurred between the start of the

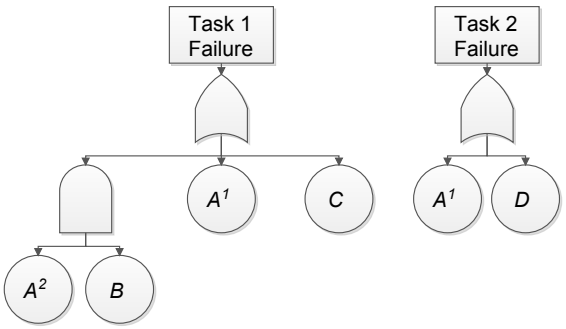

Figure 5.6.1: Fault trees representing the failure of the example system to complete two tasks

Component with Failure Mode	Failure rate (Failure per hour)
A^1	$2 * 10^{-4}$
A^2	$1 * 10^{-5}$
B	$2 * 10^{-3}$
C	$9 * 10^{-2}$
D	$2.5 * 10^{-4}$

Table 5.6.1: Component failure rate

mission and the end of phase i, $P(F_1 + F_2 + \cdots + F_i)$, can be provided by using Method 3. The lower and upper bound approximations to conditional unreliability of phase i, $P(ph_i)$, can also be derived as follows: denote $P_U(ph_i)$ and $P_L(ph_i)$ be the upper and lower bound approximation probabilities to $P(ph_i)$, and $P_U(F_1+F_2+\cdots+F_i)$ and $P_L(F_1+F_2+\cdots+F_i)$ be the upper and lower bound approximation probabilities to $P(F_1+F_2+\cdots+F_i)$, $P_L(ph_i)$ and $P_U(ph_i)$ can be calculated by according to Equation (2.5.3):

$$\begin{aligned} P_L(ph_i) &= P_L(F_1 + F_2 + \cdots + F_i) - P_U(F_1 + F_2 + \cdots + F_{i-1}) \\ P_U(ph_i) &= P_U(F_1 + F_2 + \cdots + F_i) - P_L(F_1 + F_2 + \cdots + F_{i-1}) \end{aligned} \tag{5.5.8}$$

5.6 Example

Consider a system that can perform two tasks with fault trees representing the failure of the two tasks as shown in Figure 5.6.1. All components are assumed to fail according to exponential distributions with constant failure rates as shown in Table 5.6.1. Consider that the system is requested to perform task 1 in phase 1 and task 2 in phase 2 and that the durations of each phase are 10 hours and 15 hours respectively. Using BDD model 2 described in Chapter 3 to construct BDDs, the variables are order as: $A^2_{01} < A^1_{02} < A^1_{01} < B_{01} < C_{01} < D_{02}$, and the exact BDDs for phase 1 failure, F_1, and phase 2 failure,

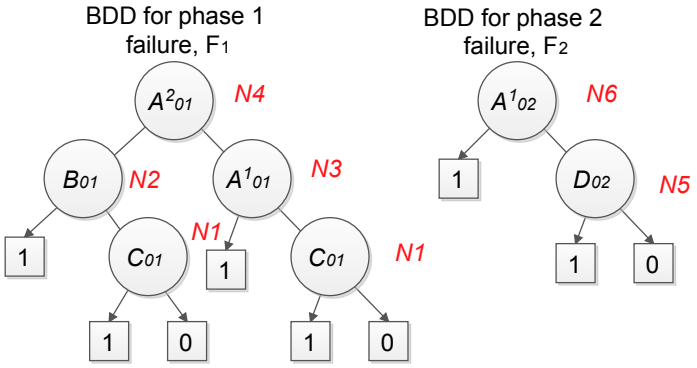

Figure 5.6.2: Exact BDDs representing phase failures

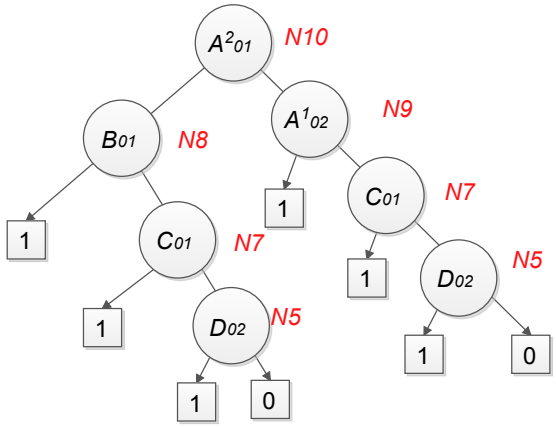

Figure 5.6.3: Exact BDD representing mission failures

F_2, are therefore as illustrated in Figure 5.6.2. The exact BDD representing $F_1 + F_2$, which represents the entire mission failure, is illustrated in Figure 5.6.3. The failure probabilities for the events in the BDD shown in Table 5.6.2 are calculated using the exponential distribution probability density function, $\int_0^t \lambda e^{-\lambda\tau} d\tau$, where t is the elapsed time since the start of the mission and λ is the one of the component failure rates shown in Table 5.6.1. Applying the quantification method for BDD Model 2 as given by Equation (3.2.2) to the BDD representing F_1 in Figure 5.6.2 and the BDD representing $F_1 + F_2$ in Figure 5.6.3, the mission failure probability Q_{miss} and the conditional unreliability of phase 1 and phase 2, $P(ph_1)$ and $P(ph_2)$, are calculated to be:

$$\begin{aligned} Q_{miss} &= P(F_1 + F_2) = 0.5980 \\ P(ph_1) &= P(F_1) = 0.5942 \\ P(ph_2) &= P(F_1 + F_2) - P(F_1) = 0.0038 \end{aligned} \tag{5.6.1}$$

Failure event	Failure probabilities
A^1_{01}	$1.9880 * 10^{-3}$
A^1_{02}	$4.9875 * 10^{-3}$
A^1_{12}	$2.9995 * 10^{-3}$
A^2_{01}	$9.9995 * 10^{-5}$
B_{01}	$1.9880 * 10^{-3}$
C_{01}	0.5934
D_{02}	$6.2305 * 10^{-3}$
D_{12}	$3.7336 * 10^{-3}$

Table 5.6.2: Component failure rate

5.6.1 Approximation Using Method 1

Applying approximation Method 1 to the BDDs representing system failure in phase 1 and phase 2, which are shown in Figure 5.6.2, the failure probability of each individual phase, $P(F_1)$ and $P(F_2)$ is calculated using the quantification method of BDD Model 2 as given in Equation (3.2.2):

$$\begin{aligned} P(F_1) &= P(N4) = 0.5942 \\ P(F_2) &= P(N6) = 0.01112 \\ Q_{miss} &= 1 - [01 - P(F_1)] \cdot [1 - P(F_2)] = 0.6396 \end{aligned} \tag{5.6.2}$$

The value of $P(F_1)$ is equal to the exact probability, since no approximation is applied. The approximation of the mission failure probability, Q_{miss}, is calculated using Equation (5.2.2). It can be seen the result obtained using Method 1 overestimates the actual mission unreliability when compared to the exact result given in Equation (5.6.1), thus Method 1 provides an upper bound approximation to the exact unreliability.

5.6.2 Approximation Using Method 2

Applying approximation Method 2 to analyse the mission requires the construction of Z-BDDs following the rules given in Equation (5.3.5). The Z-BDDs obtained for F_1 , F_2 and $F_1 + F_2$ are shown in Figure 5.6.4 and Figure 5.6.5.

The rare event approximation for mission failure using Method 2, Q_{miss}, which is a upper bound for mission unreliability, is calculated using the Z-BDD shown in Figure 5.6.5

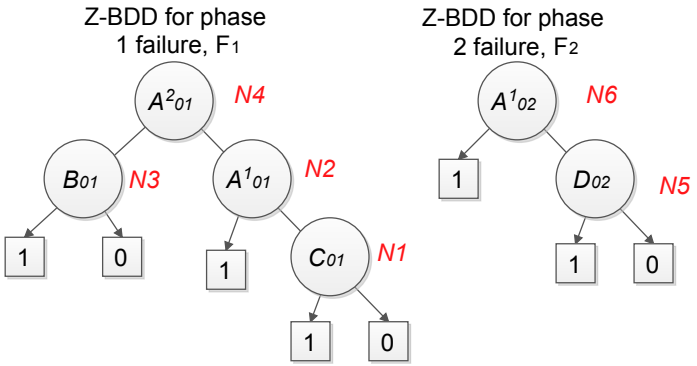

Figure 5.6.4: The Z-BDDs for failure in phase 1 and 2, F_1 and F_2

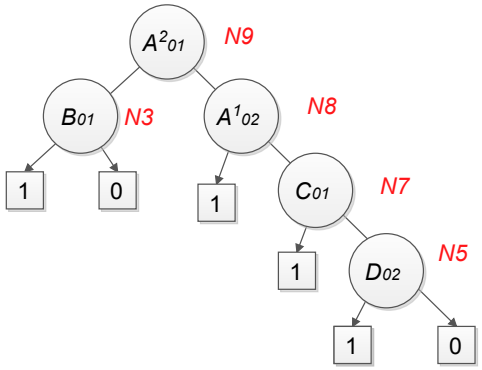

Figure 5.6.5: The Z-BDD representing mission failure

and Equation (5.3.10) by substituting the event failure probabilities in Table 5.6.2:

$$\begin{aligned} Q_{miss} &= P(N9) = p(A_{01}^2) \cdot P(N3) + P(N8) \\ &= p(A_{01}^2) \cdot p(B_{01}) + p(A_{02}^1) + p(C_{01}) + p(D_{02}) \\ &= 0.6046. \end{aligned} \tag{5.6.3}$$

5.6.3 Approximation Using Method 3

Now, consider the approximation of the mission unreliability using a truncated BDD. Firstly, an analytic truncation limit is calculated using Equation (5.5.6) to control the error between the approximate and the exact solutions for the mission unreliability. Suppose the order of magnitude of the required precision for mission unreliability is 10^{-3} (i.e. $\epsilon = 10^{-3}$). The number of paths ending at terminal 1 nodes in the BDDs representing phase 1 failure, F_1, and phase 2 failure, F_2, are 4 and 2 respectively and the number of paths ending at terminal 0 nodes in F_2 is 2. Thus, the analytical truncation limit for the system mission failure, $F_1 + F_2$, is calculated as $TL = \frac{10^{-3}}{4+2*2} = 1.25 * 10^{-4}$.

Using Algorithm 5.1, the BDD for the example phased mission is constructed as follows. Initially, $\theta_{N4+N6} =< 1, 1 >$ is assigned to the starting operation $N4 + N6$. Since for operation $< N4 + N6, \theta_{N4+N6} >$, the variable of the resultant BDD node of $N4 + N6$ is A_{01}^2 and $\gamma_{N4+N6} = 1$, the 1-branch operation $N2 + N5$ will have $\theta_{N2+N5} =< 1, A_{01}^2 >$ and the 0-branch operation $N3 + N6$ will have $\theta_{N3+N6} =< 1, \overline{A_{01}^2} >$, according to Algorithm 5.2 and Algorithm 5.3, respectively.

For operation $N2 + N5$ with $\theta_{N2+N5} =< 1, A_{01}^2 >$, the variable of the resultant BDD node is B_{01}, which relates to a different component to the component of $\gamma_{N2+N5} = A_{01}^2$. For the 1-branch operation $1 + N5$, $\theta_{1+N5} =< p(A_{01}^2), B_{01} >$. Since $p_{1+N5} = p(A_{01}^2) < 1.25 * 10^{-4}$, $1 + N5$ returns a terminal 0 node. For the 0-branch operation $N1 + N5$, $\theta_{N1+N5} =< p(A_{01}^2), \overline{B_{01}} >$. Since $p_{N1+N5} = p(A_{01}^2) < 1.25 * 10^{-4}$, $N1 + N5$ returns terminal node 0 as $1 + N5$; Therefore, the resultant BDD node for operation $N2 + N5$ is a terminal 0 node, as shown in the box in Figure 5.6.6.

For operation $N3 + N6$ with $\theta_{N3+N6} =< 1, \overline{A_{01}^2} >$, the variable of the resultant BDD node is A_{02}^1, which relates to the same component as $\gamma_{N3+N6} = \overline{A_{01}^2}$. For the 1-branch operation $1 + N3$, $\theta_{1+N3} =< 1, \overline{A_{01}^2} \cdot A_{01}^2 >=< 1, A_{01}^2 >$, the built BDD is a terminal 1 node.

For the 0-branch operation $N1+N5$ with $\theta_{N1+N5} =< 1, \overline{A_{01}^2} \cdot \overline{A_{01}^2} >$, the variable of the resultant BDD node is C_{01}, which relates to a different component than $\gamma_{N1+N5} = \overline{A_{01}^2} \cdot \overline{A_{01}^2}$.

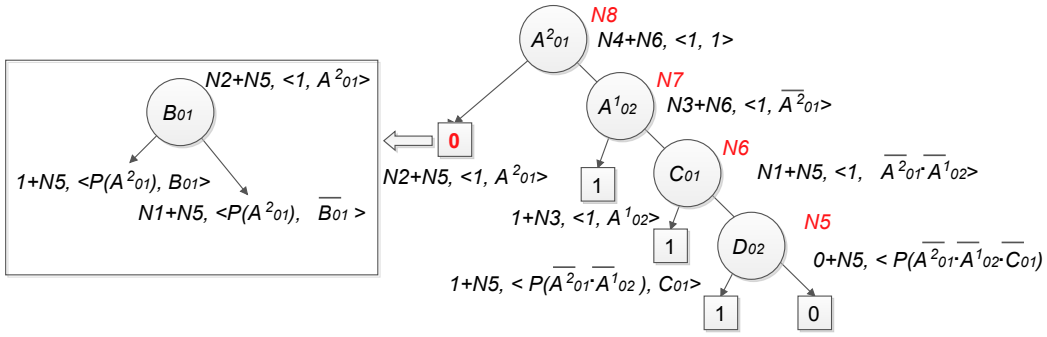

Figure 5.6.6: Truncated BDD representing the mission failure of the example system shown in Figure 5.6.1

For the 1-branch operation $1 + N5$ with $\theta_{1+N5} =< p(\overline{A_{01}^2} \cdot \overline{A_{01}^2}), C_{01} >$, the built BDD is a terminal 1 node. For the 0-branch operation $0 + N5$, $\theta_v =< p(\overline{A_{01}^2} \cdot \overline{A_{01}^2}), \overline{C_{01}} >$, the resultant BDD node is $N5$.

The truncated BDD representing mission failure for the example PMS under the analytical truncation limit $TL = 1.25 * 10^{-4}$ is shown in Figure 5.6.6.

Using the DEP-BDD evaluation method to quantify the truncated BDD in Figure 5.6.6, the mission unreliability is calculated to be 0.5979.

This is a lower bound for the mission unreliability. An upper bound for the mission unreliability can be achieved by returning a terminal 1 node for the 1-branch operation of $N8$ in Figure 5.6.6, since the upper probability of operation $N2 + N5$ is smaller than the defined truncation limit. Deploying the same quantification method used when calculating the lower bound approximation, the upper bound approximation value for the mission unreliability is calculated to be 0.5980.

5.6.4 Summary

It can be seen from this example that although Method 1 produces an upper bound, it does not offer an accurate estimation of the mission unreliability when compared with the other two methods. However, it could compute this estimation within a relatively short time, since only quantitative analysis is required after a mission configuration is defined; no other computation or construction is needed. The potential application of Method 1 is therefore quite restricted, since it can only be used as a method to approximate mission unreliability when the time available to make a decision is extremely limited and the accuracy requirement is not high.

Method 2 also provides an upper bound to the mission unreliability. Compared with Method 1, the analysis involved in applying Method 2 is more complex, as more computation is involved to construct the Z-BDD and then perform quantitative analysis. However, the accuracy achieved using method 2 is higher than that achieved using Method 1.

The exact mission failure probability is 0.5980 using BDD Model 2 in Chapter 3. The accuracy losses of the upper bound approximation using Method 1, Method 2 and Method 3 when compared with the exact mission unreliability are: 6.5%, 1.1% and 0.00% respectively. Method 3 also gives a lower bound approximation with an accuracy loss of 0.02%. The results obtained for this example indicates that the bounds produced using Method 3 are very accurate under the analytical truncation limit.

5.7 Comparison and Summary

Three approximation methods are proposed in this chapter, each of which can be used to supply information to a decision support tool for PMS in order to quickly choose the optimal alternative mission after an internal system failure or external environmental threat is detected. In both of these situations, there may not be enough time to perform exact quantitative analysis of the possible mission alternatives, if the PMS needs to make a change to its mission plan immediately, and a slow decision could have severe consequences for the security of the PMS.

Of the three approximation methods, Method 3 gives both lower and upper bound approximations to the conditional unreliability of mission phases and the entire mission unreliability while Method 1 and Method 2 give upper bound approximations of only the mission unreliability. Three measures could be used to compare the efficiency and accuracy of the developed models in terms of analysis time, memory storage and analysis accuracy:

1. The BDD size: The smaller the BDD size, the less memory is required for storage and the more efficient the method is.

2. The mission unreliability analysis time: The less the analysis time, the more efficient the method is. Since the approximations would be calculated for mission alternatives when a decision is required as to the best next course of action while a mission is underway, all analysis is performed online.

3. The entire mission unreliability approximation: The closer the approximation is to the exact mission unreliability, the more accurate the method is.

This section begins with an introduction to an indicator, which can be used to evaluate the effectiveness of the developed approximation methods accounting for the relative importance of speed and accuracy; it is then followed by an investigation into the performance of Method 3 under different truncation limit, since the choice of truncation limit can potentially impact its analysis speed and accuracy; after that, a comparison of the three developed approximation methods are made and conclusions are drawn after a number of testings have been conducted on random generated PMS using the algorithm shown in Appendix A.

5.7.1 Indicator I_λ

Method 1 and Method 2 produce upper bound approximations of the mission unreliability, and Method 3 produces both upper and lower bound approximations to the conditional unreliability of mission phases and the mission unreliability. In order to compare the mission unreliability analysis efficiency of these methods while considering the amount of accuracy loss, an indicator is created to consider the analysis time reduction and accuracy loss when using these methods to analyse mission unreliability.

The indicator can be adjusted to account for the relative importance of speed and accuracy. The indicator is expressed as:

$$I_\lambda = 0.5 * E_{ff} + \lambda * T_{red} + (0.5 - \lambda) * A_{cc}, \tag{5.7.1}$$

where

- λ is the weight parameter, which is defined according to the relative importance of the reduction in time when compared to the accuracy. λ can take a value between 0 and 0.5, with 0 meaning the time reduction is not important at all and 0.5 meaning accuracy is not important at all. A value of 0.25 means that time reduction and accuracy are considered equally important.

- E_{ff}, the effectiveness, is the percentage of missions for which the time taken to perform the analysis required to approximate the unreliability is less than the time taken to perform the exact quantification.

- T_{red}, the time reduction, measures the efficiency of the approximation method and is defined as the average percentage time reduction seen for all effective observations (i.e, those for which the time taken to approximate the unreliability is less than that

taken to calculate exactly) when compared to the exact quantification time, i.e.,

$$T_{red} = \frac{T_{exact} - T_{app}}{T_{exact}}, \tag{5.7.2}$$

where T_{app} is the time needed to approximate the lower or upper bound to the mission unreliability and T_{exact} is the time taken to quantify the exact mission unreliability.

- A_{cc}, the accuracy, measures the accuracy of the approximation of the mission unreliability and is defined as the approximation as a percentage of the exact value for lower bound and the exact value as a percentage of the approximation for upper bound:

$$A_{cc} = \begin{cases} \frac{U_{app}}{U_{exact}} & \text{Lower bound approximation} \\ \frac{U_{exact}}{U_{app}} & \text{Upper bound approximation,} \end{cases} \tag{5.7.3}$$

where U_{app} is the lower or upper bound to the approximated mission unreliability probability and U_{exact} is the exact mission unreliability.

Since the value of E_{ff} is between 0 to 1; the time reduction is at least 0; the accuracy is between 0 and 1; the value of all the parameters are positive, the bigger the value of E_{ff}, T_{red}, A_{cc}, the bigger the value of I_λ and hence the more desirable the approximation method is.

The indicator I_λ is then used to select the truncation limit that leads to the most efficient performance of Method 3 under different requirements for analysis speed and accuracy. Then, it is used to compare the efficiency of the two valid approximation methods, Method 2 and Method 3, after eliminating Method 1 due to its poor accuracy approach.

5.7.2 Truncation Limit Investigation in Method 3

The size of a truncated BDD constructed using Method 3 depends on the choice of the truncation limit TL, i.e., the smaller the TL, the bigger the BDD size and longer the analysis time required. Although using the analytical truncation limit can ensure an accurate approximation is achieved, for cases when the truncation limit is too small, Method 3 could be ineffective, meaning that the time needed to approximate the mission unreliability will be longer than the time needed to perform the exact analysis using the BDD Model. Therefore, before comparing the three methods, an investigation into the effect of the truncation limit is conducted in this section.

5.7.2.1 Example Illustration

Method 3 gives a lower bound approximation to the mission unreliability by cancelling the operation of certain BDD nodes and returning terminal 0 nodes. The following example compares of the mission unreliability analysis time, mission unreliability accuracy and size of BDD analysed using the exact BDD model and Method 3.

Consider a random generated 7-phase PMS using the random fault tree generator in in Appendix A. The lower bound approximations to conditional unreliability of each mission phase and the mission unreliability with increasing truncation limit (TL) from 10^{-15} to $10^{-0.5}$ are shown in Table B.3.6, where column 1 shows the TL increasing from 10^{-15} to $10^{-0.5}$ in steps of $10^{0.5}$. For each TL, column 2 to column 8 show the analysis time for unreliability of mission phase 2 to mission phase 7 and column 9 shows the analysis time for the entire mission unreliability. Column 10 and Column 11 show the approximate mission unreliability and the size of the truncated BDD representing mission failure given the considered TL. The exact unreliability of mission phases and the whole mission, the sizes of the BDDs representing the conditional phase failures and entire mission failure, and the time needed to analyse these BDDs for the PMS considered are shown in Table 5.7.1.

	Phase 1	Phase 2	Phase 3	Phase 4	Phase 5	Phase6	Phase7	Mission
BDD size	465	4363	4435	4557	4694	4850	4944	4944
Analysis Time(s)	0.04	0.06	0.19	0.07	0.08	0.21	0.14	0.79
Unreliability	2.71E-29	4.13E-11	5.19E-09	6.96E-06	1.90E-4	4.52E-5	0.31	0.31

Table 5.7.1: Exact analysis data for example mission

The impact of the choice of the truncation limit on the considered approximation measurement values (approximation probabilities, BDD size and time) is shown in Figure 5.7.1 to Figure 5.7.4.

Figure 5.7.1 shows that for each individual phase, the analysis time decreases with increasing TL. If for certain TL value, a phase takes the longest time to analyse its unreliability, it would probably also take the longest analysis time for each TL value considered.

Figure 5.7.4 compares the difference between the approximated lower bound probability and the exact mission unreliability and demonstrates that although there is a relatively large increase in the difference when the TL increase from $10^{-5.5}$ to 10^{-5}, the lower bound provided by Method 3 is still very close to the exact mission unreliability in all cases. This is clear since the error is smaller than 10^{-4}, which can be seen from the y-axis.

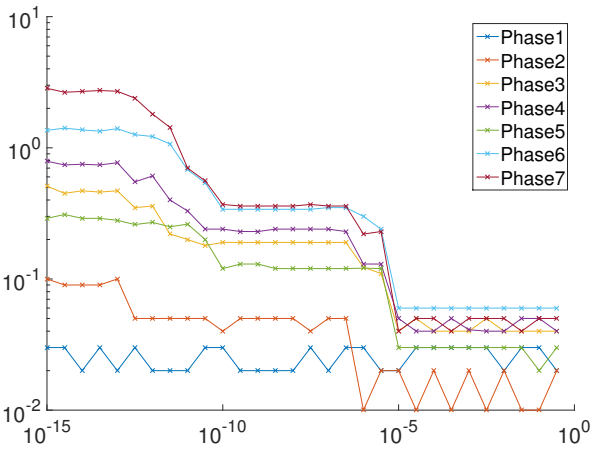

Figure 5.7.1: Trend of phase analysis time with increasing TL

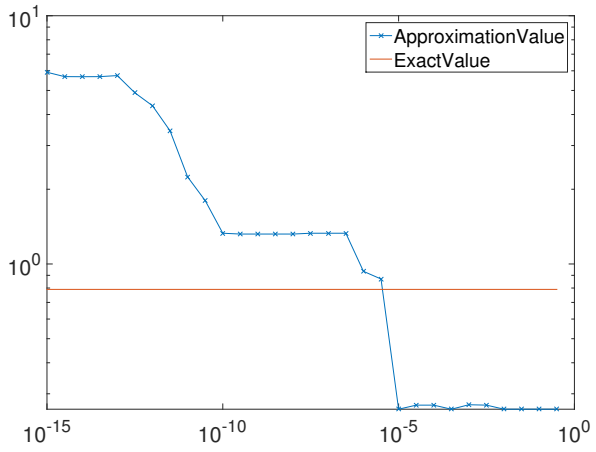

Figure 5.7.2: Trend of mission analysis time with increasing TL

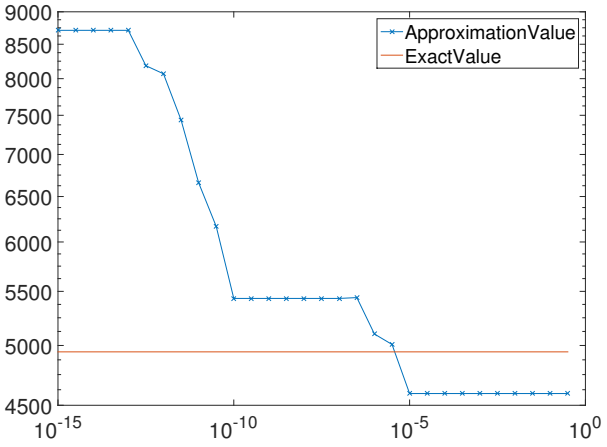

Figure 5.7.3: Trend of mission BDDs sizes with increasing TL

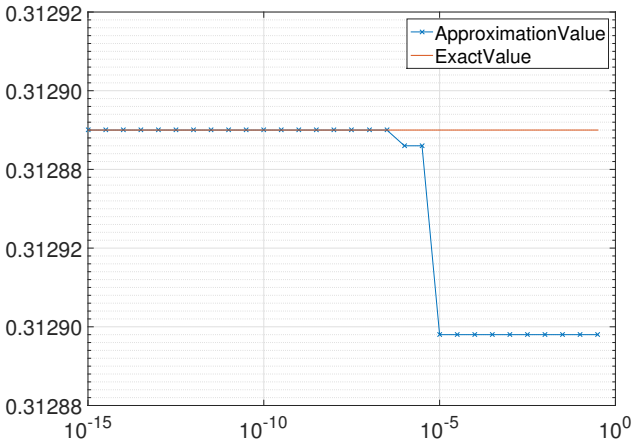

Figure 5.7.4: Trend of mission unreliability with increasing TL

Comparing all of the graphs, similar drops in the value of the measures (individual phase unreliability analysis time, mission unreliability analysis time, size of the BDD representing mission failure, and the approximation of the mission unreliability), can be seen when using Method 3 and the TL drops from $10^{-5.5}$ to 10^{-5}.

These results demonstrate that approximation Method 3 can be used to calculate a lower bound approximation to the mission unreliability more rapidly than analysis using the exact BDD model without significant loss in accuracy. For the same mission, the bigger the selected truncation limit, the more rapid the approximation analysis can be performed.

5.7.2.2 Choice of TL Under Different Requirements for Accuracy

To investigate the effect of TL choice on the approximation of the entire mission unreliability using Method 3, tests are conducted on the 100 benchmark PMS so that an appropriate TL can be chosen when there is different requirement for the importance of analysis speed and accuracy for Method 3.

Table B.3.1, Table B.3.2, and Table B.3.3 show the percentage of time reduced when using Method 3 to compute the lower, upper and both bounds approximation probabilities to the mission unreliability given different truncation limits, using Equation (5.7.2), for each PMS.

In Table B.3.1, columns 2 to 10 show the percentage reduction in time taken to compute the lower bound approximation of the unreliability for the entire mission given varying truncation limit. The time reduction for all tested systems under the analytical truncation

limit and truncation limits varying from 10^{-15} to 10^{-1} are, on average, 42.49%, 41.48%, 31.42%, 31.79%, 45.44%, 56.27%, 71.51%, 80.86%, 80.64%, respectively. When $TL \geq 10^{-3}$, the percentage reduction in time on average is the closest to 100%, meaning that great analysis speed advantage can be achieved using Method 3 and these TL. Table B.3.2 and Table B.3.3 have the same structures as Table B.3.1 to display the percentage reduction in time using Method 3. Table B.3.2 shows when $TL = 10^{-1}$, the average reduction in time taken to approximate the upper bound of the mission unreliability is closest to 100% and Table B.3.3 shows when $TL = 10^{-3}$, the average reduction in time taken to approximate both the lower and upper bounds of the mission unreliability is closest to 100%.

Table B.3.4 and Table B.3.5 show the accuracy of the lower and upper bound approximations compared to the exact mission unreliability given different truncation limits, calculated using Equation (5.7.3). In Table B.3.4, columns 2 to 10 show the accuracy of the approximated lower bound compares to the exact mission unreliability under varying truncation limits. The accuracy for all tested example systems, on average, under the analytic truncation limit and those varying from 10^{-15} to 10^{-1} are 100.00%, 100.00%, 99.98%, 99.88%, 99.77%, 99.02%, 94.02%, 80.91%, and 30.18%, respectively. When TL is analytical or $TL \leq 10^{-7}$, the accuracy of the lower bound is almost 100%, which means on average, there is less than 1% loss in accuracy. When $TL \leq 10^{-3}$, the average accuracy is greater than 80.91%. If we assume the reasonable accuracy is 80%, then, the lower bound mission unreliability approximation produced by method 3 is reasonably accurate when $TL \leq 10^{-3}$.

Table B.3.5 has the same structure as Table B.3.4 to display the accuracy given different truncation limits when calculating an upper bound. When the TL is analytical or $\leq 10^{-7}$, the average accuracy is over 88.30%. The accuracy of the approximated mission unreliability upper bounds drop significantly when $TL \geq 10^{-5}$. Thus, the upper bounds mission given by Method 3 are reasonably accurate when $TL \leq 10^{-7}$ and when $TL \geq 10^{-5}$, the upper bound approximation should be used with care.

Although the accuracy and time taken to perform the approximation each gives a good indication of the performance of the approximation method, it might be useful to measure the overall efficiency of the method. To do this, the indicator I_λ can be used.

Using the values in Table B.3.1 to Table B.3.4 and Equation (5.7.1), the value of the indicator I_λ for different values of λ, $I_{0.05}$, $I_{0.1}$, $I_{0.3}$, and $I_{0.4}$, and different truncation

limits are shown in Table 5.7.2, where I_λ (L) shows the performance efficiency of the lower bound approximation of the mission unreliability using Method 3 and I_λ (U) shows the performance efficiency of the upper bound approximation of the mission unreliability using Method 3.

In all cases, the bigger the value of I_λ, the better Method 3 is using the given truncation limit. When approximating the lower bounds to mission unreliability, Method 3 is the most efficient given truncation limit 10^{-1} when evaluated using indication $I_{0.05}$, $I_{0.1}$, $I_{0.3}$, and $I_{0.4}$. When approximating the upper bounds to mission unreliability, Method 3 is the most efficient given truncation limit 10^{-1} when evaluated using indication $I_{0.3}$ and $I_{0.4}$ and it is the most efficient given truncation limit 10^{-9} when evaluated using indication $I_{0.05}$, $I_{0.1}$,

Therefore, for the lower bound approximation using Method 3, $TL = 10^{-1}$ should be always chosen. For the upper bound approximation, when the accuracy requirement is high, $I_{0.05}$ will be chosen as the performance indicator, and $TL = 10^{-9}$ should therefore be chosen in order to conduct the approximation analysis. When time saving is more important, $I_{0.4}$ will be chosen as the performance indicator, and $TL = 10^{-1}$ should therefore be chosen to construct truncated BDDs and perform the approximation. For analysis where there exists no significant preference for time saving over accuracy or vice-versa, $I_{0.3}$ will be chosen as the performance indicator, in which case the optimal truncation limit is $TL = 10^{-1}$.

	Analytic TL	10^{-15}	10^{-13}	10^{-11}	10^{-09}	10^{-07}	10^{-05}	10^{-03}	10^{-01}
$I_{0.05}$ (L)	0.955	0.944	0.955	0.941	0.932	0.921	0.936	0.929	0.993
$I_{0.05}$ (U)	0.843	0.876	0.863	0.877	0.886	0.868	0.875	0.870	0.739
$I_{0.1}$ (L)	0.911	0.923	0.934	0.918	0.911	0.901	0.914	0.905	0.990
$I_{0.1}$ (U)	0.815	0.871	0.862	0.870	0.881	0.864	0.871	0.866	0.766
$I_{0.3}$ (L)	0.732	0.840	0.852	0.830	0.829	0.822	0.826	0.811	0.979
$I_{0.3}$ (U)	0.700	0.849	0.857	0.840	0.859	0.849	0.853	0.851	0.874
$I_{0.4}$ (L)	0.643	0.799	0.811	0.785	0.788	0.782	0.782	0.764	0.974
$I_{0.4}$ (U)	0.642	0.838	0.855	0.826	0.848	0.841	0.845	0.843	0.928

Table 5.7.2: Comparison of choice of TL on the impact of the approximation analysis of Method 3 with varying I_λ

5.7.3 Comparison of the Three Truncation Methods

Method 1, Method 2 and Method 3 (here used to give an upper bound approximation only in order to allow comparison with the other two methods) are first tested on 5 sample PMS in order to see if there is a clear difference in performance between the three methods, which would mean that any of them might be discounted.

The mission unreliability upper bound, analysis time and size of the BDD representing failure of the entire mission using each of the approximation methods and the exact BDD analysis for each PMS is shown in Table B.3.7 and the A_{cc} and T_{red} values calculated according to Equation (5.7.2) and Equation (5.7.3) using each approximation method for each PMS are shown in Table 5.7.1.

In Table B.3.7, columns 2 to 5 show the upper bound approximations of the mission unreliability using Method 1, Method 2, Method 3, and the exact BDD model respectively; columns 6 to 9 show the mission unreliability analysis time using Method 1, Method 2, Method 3, and the exact BDD model; the last three columns show the sizes of BDDs representing the mission failure obtained using method 2, method 3, and exact BDD model respectively. It can be seen that the upper bound approximation by Method 1 is quite inaccurate, i.e., when the exact mission unreliability is close to 0, the approximation probability is almost 1. The inaccuracy of Method 1 is also demonstrated by the values in column 2 in Table 5.7.3, and the average accuracy is only 0.20%. Although plenty of time (almost 99% for all cases) can be saved by using Method 1, it is not acceptable for use, since the accuracy loss is too great, for example in the third mission considered, Method 1 only provides 0.03% accuracy. The overestimate of the mission unreliability using Method 1 could cause unnecessary termination of a mission and eliminate possible alternative mission configurations if Method 1 is used to provide information for use in a decision making process.

Analysis using Method 2 and Method 3 can save as much as 90% of the analysis time with reasonable accuracy loss when compared to the exact analysis. In order to compare the efficiency of these two more practical approximation methods, more tests are carried out using approximation Method 2 on the same PMS tested using Method 3 in Section 5.7.2.2. The results are shown in Table B.3.8. In 92 out of 100 cases, Method 2 is faster than the exact analysis and thus the effectiveness of Method 2 is 92%. The percentage of reduction in time using Method 2 compared to the exact analysis is 72.29%, while the average accuracy is 80.68%.

Four different performance indicators, $I_{0.05}$, $I_{0.1}$, $I_{0.3}$, and $I_{0.4}$, which can be used to compare the efficiency of approximation methods given different analysis time and accuracy requirements are now considered in order to decide under what importances each method should be used as a decision support tool for system performing phased missions for which decisions are needed to the best next course of action.

	Accuracy approach			Time saved		
observation	M1	M2	M3($TL = 10^{-5}$)	M1	M2	M3($TL = 10^{-5}$)
1	0.03%	83.87%	20.47%	97.08%	93.96%	72.81%
2	0.52%	93.59%	100.00%	99.86%	99.52%	87.71%
3	0.03%	90.63%	77.80%	99.71%	98.97%	66.25%
4	0.33%	94.34%	93.36%	99.97%	99.90%	91.08%
5	0.07%	85.61%	62.55%	99.47%	98.54%	91.35%
Average	0.20%	89.61%	70.84%	99.22%	98.18%	81.84%

Table 5.7.3: Analysis data for example missions using different upper bound approximation methods

Using Equation (5.7.1), values of the four indicators are calculated for Method 2 and shown in the second column of Table 5.7.4. These are compared to the same indicator values that were calculated for Method 3 in Section 5.7.2.2, with truncation limits giving the highest indicator values chosen for the comparison (10^{-9} for $I_{0.05}$ and $I_{0.1}$, and 10^{-1} for $I_{0.3}$ and $I_{0.4}$). The values for Method 3 are shown in the third column of Table 5.7.4.

The table demonstrates that for each I_λ considered, the indicator value of Method 3 is always greater than that of Method 2, meaning that Method 3 always performs better than Method 2 even as the requirement for analysis speed or accuracy varies. When $\lambda = 0.05, 0.1, 0.3$, the performances of Method 2 and Method 3 are relatively close while when $\lambda = 0.4$, i.e., time reduction (analysis speed) is considered to be more important than accuracy, Method 3 is much better than Method 2.

	Method 2	Method 3(Max)
$I_{0.05}$ (U)	0.859	0.886
$I_{0.1}$ (U)	0.855	0.881
$I_{0.3}$ (U)	0.838	0.874
$I_{0.4}$ (U)	0.830	0.928

Table 5.7.4: Comparison of method 2 and method 3

5.7.4 Summary

During a mission, the time available to make a decision as to the best next course of action can be limited in urgent situations, in which case, approximations must be used in order to obtain the updated failure probabilities for alternative mission configurations in the shortest possible time with reasonable accuracy loss. This chapter has developed three approximation methods based on BDD models. Method 1 and Method 2 give upper bounds of the mission unreliability while Method 3 could provide both lower and upper bounds

to the conditional unreliability of mission phases and the entire mission unreliability.

Method 1 is developed based on the mission unreliability upper bounds presented in [12][46][47] and uses the BDD approach to compute individual phase unreliability rather than the traditional inclusion-exclusion expansion method, which requires minimal cut set computation. Since Method 1 involves converting task failure fault trees into BDDs in the off-line stage of the decision making process presented in Section 2.6, only quantification of the BDDs is needed in the online stage and thus the analysis speed of Method 1 is very fast. However, the tested results in Table 5.7.3 also reveal that the accuracy loss of Method 1 is very high (the accuracy is only 0.20%), which makes Method 1 impractical for real-time analysis within a decision support tool for autonomous systems.

Method 2 is a rare event approximation computed using Z-BDDs. Z-BDD construction rules and quantification for PMS containing multiple failure mode components are developed based on Shannon's decomposition for coherent systems [17]. The results in Table B.3.8 demonstrate that by using Method 2, a great improvement in analysis time (time reduction of 72.99%) with reasonable accuracy loss (accuracy of 80.68%) can be achieved when approximating the mission unreliability for coherent systems.

Method 3 provides approximations to the conditional unreliability of mission phases and the entire mission unreliability using truncated BDDs. An algorithm to construct truncated BDDs while considering dependencies arising due to phases and multiple failure mode components is developed. Since the sizes of truncated BDDs depend on the value of selected truncation limit, an analytical technique is developed in order to find a truncation limit that can be used to control the accuracy loss of the lower bound approximation provided by Method 3 compared to the exact mission unreliability. An indicator I_λ with varying value of λ is created in order to help choosing the truncation limit when there are different requirements for analysis speed and accuracy. The results shown in Table 5.7.2 illustrate that for lower bound approximations, $TL = 10^{-1}$ is the optimal truncation limit to choose to perform Method 3 analysis and for upper bound approximations, $TL = 10^{-09}$ should be chosen when the accuracy requirement for Method 3 is high, otherwise, $TL = 10^{-01}$ is selected, since these truncation limits give the highest I_λ values.

On average, all of the approximation methods need less analysis time than the exact BDD analysis when used to calculate the upper bound approximations to the mission unreliability. If the calculated upper bound approximation is acceptable the analysed mission configuration could be considered an acceptable alternative to a current mission,

which is no longer considered safe due to the detection of faults. The approximation methods can also be used to facilitate the analysis of as many alternative missions as possible in a short time in order to help to choose the optimal mission configuration for a PMS to perform when a change of mission is required.

Of the three developed approximation methods, Method 3 appear to be the most promising, since it can offer not only lower and upper bound approximations to the entire mission unreliability, but also to the conditional unreliability of mission phases. The extra approximation probabilities provided by Method 3 can be more helpful when used in urgent decision making situations. If the lower bound of the approximated mission unreliability is still too high to accept, the considered mission could be eliminated as an alternative mission. The approximation to the conditional unreliability of mission phases could be used to help the PMS to choose the best next course of action in the shortest possible time, since the time needed to analyse the next phase should be shorter than that required to analyse the whole mission. Therefore, when a decision is needed instantly, the success of only the next task might be considered in order to decide what to do next, giving more time to analyse other mission scenarios to decide the entire mission profile.

Chapter 6

Case Study

6.1 Introduction

The reliability analysis methods considered in the course of this research have been developed in the context of their application in supporting decision making within autonomous systems. The purpose of the research has been to develop methods of analysis that can be applied to evaluate the probability of mission failure as accurately as possible, in the shortest time possible, so that information can be provided to a decision maker (an autonomous or a human decision maker), which can then decide the best next course of action for a system following an event such as an internal system failure or the emergence of an external threat. For systems such as UAV, the ability to rapidly respond to emerging events is particularly important, since they commonly operate in environments where the failure to act quickly may lead to consequences of varying severity, ranging from possibilities such as the loss of an expensive platform to the loss of human life, depending on the area in which the platform is operating.

When an event that has the potential to affect the success of a mission is found to have occurred, reliability analysis can be carried out to determine the probability of failure of the system for the remainder of the mission. If this is unacceptable, other mission configurations must be considered and a decision made as to which mission configuration should be adopted by the system in order to offer the greatest probability of success. If a decision must be made quickly then fast, accurate reliability analysis is crucial in order to be able to consider as many alternative missions as possible in the time available to make a decision and to ensure a sufficient degree of confidence in the failure probabilities obtained so that the decision made is based on sound information.

An example system is considered in this chapter in order to demonstrate the application of reliability analysis as a decision support tool for an autonomous system. A UAV is assumed to be taking part in a search and rescue (SAR) mission and the BDD models and analysis techniques considered throughout this research are applied in a number of situations both before and during the mission in order to demonstrate their potential impact in each of the situations. Consideration is given to the amount of offline and online analysis that must be performed and what implication this has in the time taken to provide the information required to support decision making.

The purpose of this chapter is to shown how real-time decisions are made by the PMS under different scenarios based on the time taken to calculate updated phase and mission unreliabilities and their accuracy (when approximation methods are employed). Before a mission starts, the investigated BDD models and ordering schemes are used to calculate the conditional phase and mission unreliabilities to decide whether the proposed mission configuration can be safely carried out. The BDD model that leads to the smallest BDD sizes is then used for future updated unreliability calculations. While the original mission is being performed, if failures are detected that lead the calculated updated mission unreliabilities to be unacceptable, then it is considered that it is not safe to continue with the original mission and alternative mission configurations need to be considered. The BDD model and ordering schemes that showed most promise for calculating results quickly and accurately (as discussed Chapter 3 and Chapter 4) are used to calculate the conditional phase and mission unreliabilities of alternative mission configurations. The efficiency advantage of the BDD model and the ordering schemes that leads to the least analysis time can then be demonstrated by comparing the analysis time using the other BDD models and ordering schemes. When the time available to make a decision is limited (i.e., under emergency situations like avoiding a storm nearby), the three approximation methods (as discussed in Chapter 5) are adopted under varying importance of speed and accuracy. The efficiency advantage using the three developed approximation methods over the exact BDD analysis is demonstrated by comparing the time taken to analyse alternative mission configurations.

By combining the most promising methods developed in each of the previous chapters, and applying them to an example system which contains features of a real world platform, the efficiency advantage of the developed methods can then be demonstrated within a single application. By applying the methods to the example system, it also demonstrates how

the developed methods can be used to support decision making process for autonomous systems performing phased missions, as they encounter different scenarios.

6.2 Mission Description

Consider a UAV which can perform 7 different tasks being used in a SAR mission. The fault trees representing the failures of these tasks are denoted as T1 to T7. The fault trees are generated using the random fault tree generator in Appendix A with parameters as follows: maximum number of components 30, maximum number of inputs of each fault tree layer 10, percentage of gate inputs in each fault tree layer $a = 0.9/b = 0.1$, maximum number of component failure modes 3, and percentage of multiple failure mode components 0.2. The phases of the original mission are shown in Table 6.2.1.

Phase	1	2	3	4	5	6
Fault Tree	T1	T2	T3	T4	T5	T6

Table 6.2.1: The example SAR mission phases

Before the SAR mission starts, the UAV must perform reliability analysis on the mission defined in Table 6.2.1 based on current platform and environmental conditions. At this early stage, it is assumed that there is enough time available to calculate the mission unreliability and thus BDD models that gives the exact mission unreliability are used to conduct the analysis. If the calculated unreliability is too high to accept, then another mission configuration must be considered as illustrated in Figure 2.6.1.

The constructed BDDs for each task are stored in the system and will be used during an online analysis if any change is detected and the BDDs will be quantified again with updated event failure probabilities to check whether the unreliabilities are still acceptable. For this reason, the smaller the stored BDDs, the faster the quantification process will be. Thus, the global ordering scheme and the BDD model that lead to the smallest BDD sizes should be used. Also, due to the efficiency advantage of the proposed BOI scheme on analysing the unreliability of the first altered phase in an alternative mission as demonstrated in Chapter 4, the BDDs constructed using the BOI scheme are also stored in the system when memory space is available so that the conditional unreliabilities of the first altered phases in the mission alternatives can be rapidly calculated to help the UAV making a sound decision.

All of the above computation can be conducted offline and will not need to be performed

again when a decision is required, meaning that the time required for online analysis is minimised.

After the SAR mission has started, in order to ensure the UAV can quickly respond to the changing environment so that the SAR mission can be completed successfully, the updated unreliabilities of the current and possible alternative missions need to be calculated in the shortest possible time whenever new information is provided by the UAV diagnostic tool. For example, if bad weather has been detected which would affect the performance of the PMS in a certain phase and will lead to the current mission being unsafe to continue, then alternative mission configurations need to be considered to ensure the search and rescue objective can still be achieved. Since the environmental change can most likely be predicted in advance, it will most likely not lead to an urgent situation that needs almost immediate decisions to be made. Therefore, exact BDD models can be used to perform the updated mission unreliability analysis or the updated conditional unreliability analysis of the next phases for alternative missions. If a serious system failure, such as an engine fault, is detected, then this means the time available for the UAV to make a decision as to the best next course of action could be quite limited, meaning that the approximation methods developed in Chapter 5 must be used to in order to provide a much faster reliability analysis for possible mission alternatives with reasonable accuracy loss.

6.3 Before the Mission Starts: Comparison of BDD Models

6.3.1 Introduction

Before the SAR mission starts, exact mission unreliability for the mission described in Table 6.2.1 must be performed to help the UAV make a decision as to whether to follow a proposed mission configuration to achieve the mission objective based on the current system conditions. If each conditional phase unreliability and the mission unreliability are at an acceptable level, the mission will start, otherwise, an alternative mission configuration will be considered. All ordering schemes and BDD models that provide the exact mission unreliability are tested in order to choose ones that lead to the smallest size of BDD representing the mission failure and the least mission unreliability analysis time. All of the computation performed before mission starts is offline and thus there is enough time to select the optimal BDD model and variable ordering scheme that have the

highest potential to provide the most efficient analysis of the updated mission unreliability when the mission is underway. Some of the analysis results acquired offline, for example, the variable ordering list and the BDDs constructed using the BOI scheme, can be used for updated unreliability analysis for alternative mission configurations when any system condition change is detected while the mission is underway without any extra time cost for the online analysis.

6.3.2 Comparison of BDD Models and Ordering Schemes

The fault trees must be converted to BDDs using the most efficient variable ordering scheme. Comparisons of the converted BDD size using different variable ordering schemes and different BDD models are shown in Table 6.3.1. Columns 2-5 shows the BDDs sizes and columns 6-9 show the mission analysis time using BDD Model 1 to BDD Model 4 under the nine variable ordering schemes. All of the 4 models calculate the exact mission unreliability. The Forward-BDD Model is an existing model developed in [35] whilst the other three models are newly-developed in this research project.

For the UAV system considered, the results in Table 6.3.1 show that under certain variable ordering (comparing the horizontal data), the size of BDD constructed using Model 3 and the Forward-BDD model are smaller than those constructed using Model 1 and Model 2. This result is supported by the testing results and conclusions drawn in Chapter 3. Model 3 and the Forward-BDD model use the same construction rules that were developed in [36] while Model 1 and Model 2 use the modified DEP-BDD construction rules developed in Chapter 3. The mission unreliability analysis time using the developed models, Model 1, 2 and 3 is always faster than when using the existing Forward-BDD model. Model 3 needs the least time due to the advantage of the proposed quantification method.

When considering the application of different ordering schemes for each BDD model (comparing the vertical data), Scheme 5 always leads to smallest mission failure BDD size and the least mission unreliability analysis time. The BDD size and the analysis time is consistent, i.e, smaller BDD size always implies less analysis time. The optimality of the different variable ordering schemes is consistent no matter which BDD model is used, i.e, if Scheme 5 performs the best using BDD model 3, it will also perform the best using BDD model 1, compared with the performance of the other schemes. This means the choice of BDD model will not affect the performance order of the variable ordering schemes.

Parameter	BDD size				Mission analysis time(s)			
	Model 1	Model 2	Model 3	Forward-BDD Model	Model 1	Model 2	Model 3	Forward-BDD Model
Scheme 1	5688	4860	3339	3339	2.0	1.4	0.6	10.7
Scheme 2	8309	8913	6875	6875	5.4	6.1	3.4	52.3
Scheme 3	10863	10348	6492	6492	7.4	6.6	2.6	44.3
Scheme 4	6989	6654	3928	3928	3.1	2.8	0.8	9.6
Scheme 5	3688	3300	2358	2358	0.8	0.7	0.3	4.5
Scheme 6	4754	4417	2951	2951	1.4	1.2	0.4	5.6
Scheme 7	6709	6748	4939	4939	3.0	3.1	1.5	29.9
Scheme 8	6866	6274	5397	5397	2.8	2.4	1.7	44.9
BOI	5816	5422	4820	4820	2.8	2.3	1.5	30.3

Table 6.3.1: Analysis results for original mission configuration in the off-line stage

Suppose the analysed conditional unreliability of all mission phases and the entire mission unreliability is acceptable. Scheme 5 and BDD model 3 will be selected to perform updated mission unreliability analysis after the mission has started, since they lead to the smallest mission failure BDD sizes and the least analysis time for this SAR mission. When enough memory space is available, the BDD structures constructed using the BOI scheme and Model 3 are also stored in the system in order to allow quick calculation of the conditional unreliabilities of the first altered phases of all possible mission alternatives when the current mission can no longer be safely continued and other mission configurations need to be considered.

6.3.3 Summary

All four of the investigated BDD models generate the same mission and conditional mission phase unreliabilities, which are exact probabilities. Model 1, Model 2 and Model 3 are BDD models developed in this research while the Forward BDD Model is an existing model developed in [36]. The performance of the BDD models changes consistently with the variable ordering scheme, meaning that the BDD model that leads to the fastest analysis using one variable ordering scheme will probably also lead to the fastest analysis using another scheme. All of the BDD models developed in this research showed great analysis speed improvement compared to the Forward-BDD model: Model 1 and Model 2 correct the imperfections in the DEP-BDD analysis identified in [36], which otherwise leads to inaccurate results whilst Model 3 introduced a novel quantification method which expedites the computation process for the Forward-BDD Model. As illustrated in the SAR mission, BDD Model 3 and Scheme 5 lead to the fastest analysis, this is supported by the conclusions drawn in Chapter 3 and Chapter 4. Method 3 and Scheme 5 would therefore would appear to be the most promising for use when analysing the mission unreliability of mission alternatives to help the UAV to make a decision regarding the optimal alternative

mission configuration after the SAR mission has started and new information is provided by diagnostic tool in the UAV.

6.4 Updated Mission Analysis When the SAR Mission is Underway: Comparison of Ordering Schemes

6.4.1 Introduction

During the performance of phase 2 of the SAR mission, bad weather has been detected in a future phase, for instance, the area where phase 3 will be performed. This means the probabilities of some basic events in the fault trees representing the phase and mission failures will have to be updated from a small amount to 1 or close to 1. After quantitative analysis using these updated basic event failure probabilities on the stored BDDs structures acquired during the off-line stage, the original mission unreliability is proved to be unacceptable. Therefore, the original mission configuration is no longer safe and the UAV needs to alter to a new mission configuration in order to accomplish its required objective with a lower probability of failure.

The analysis performed at this stage is online, meaning that any time needed for the updated unreliability analysis will eat into the time available for the UAV to make a decision about its future.

Since under the original mission configuration, Model 3 results in the smallest BDD size and the lowest mission analysis time, it is used to analyse the updated mission unreliability. The speed of the analysis then depends solely on the performance of the variable ordering schemes.

When a decision is not urgent (the bad weather is happening at the end of phase 3 of the SAR mission), exact calculation of the conditional unreliability of phases and the mission unreliability of possible mission alternatives will be conducted. In this case, Scheme 5 has been proved to be the most promising to provide the fastest updated reliability analysis (as seen from Table 6.3.1, it always leads to the smallest BDD size and the least analysis time when using different BDD models). When the time available to make a decision is limited (if the bad weather is happening at the beginning of phase 3 of the SAR mission), only the exact analysis of condition unreliabilities of the first altered phases in possible mission alternatives would be performed in order to make a decision as to what to do next, with a decision made later about the remainder of the mission. In this case, the stored

BDDs constructed using the BOI scheme can be deployed to provide a faster analysis than the other schemes, since the re-ordering of variables, analysis of previous phases shared by the current and the alternative missions has been performed offline and thus will not contribute to the time taken for the online analysis.

6.4.2 Advantage of the BOI Scheme

Suppose the possible mission alternatives include a new task which is not included in the original mission and the tasks in later phases and failure criteria change (variable reordering is therefore compulsory). Two scenarios are considered:

Scenario 1: The mission needs to alter from the start of the third phase, with the alternative mission configuration shown in Table 6.4.1.

Phase	1	2	3	4	5	6	7
Fault Tree	T1	T2	T7	T3	T4	T5	T6

Table 6.4.1: Mission alternative 1 task fault tree description

Scenario 2: The mission needs to alter from the start of the penultimate phase, with the alternative mission configuration shown in Table 6.4.2.

Phase	1	2	3	4	5	6	7
Fault Tree	T1	T2	T3	T4	T5	T7	T6

Table 6.4.2: Mission alternative 2 task fault tree description

The sizes of the BDDs representing a failure occurring between the beginning of the mission and the end of the first altered phase ($F_1+F_2+F_3$ and $F_1+\cdots+F_6$) and the failure of the entire mission for each alternative mission configuration are shown in columns 2-5 in Table 6.4.3 and the time to analyse the conditional unreliability of the first altered phase and the entire alternative mission unreliability is shown in columns 6-9 respectively. It is seen that the analysis speed advantage of the BOI scheme stands out in comparison to the others as shown in the last row and last four column of Table 6.4.3.

This is because variable ordering under the BOI scheme remains the same no matter how the mission is configured. The scheme sorts all possible components that can appear in all possible missions. The variable order list thus depends solely on the structures of individual task fault trees rather than mission failure fault trees. Since the BDDs for the original mission under the BOI scheme are stored in the system, when analysing the

alternative mission configurations under the BOI scheme, the time needed for re-ordering variables and constructing BDDs for previous phases can be omitted from the online analysis. The advantage is extremely significant for conditional unreliability calculation for the first altered phase in the alternative mission, i.e phase 3 in scenario 2 and phase 6 in scenario 3.

The BOI scheme may not lead to the smallest size of BDD representing the failure of the alternative mission, as illustrated in column 2-5 of Table 6.4.3, it is the optimal choice if the current mission is no longer considered safe and exact conditional unreliabilities of the first altered phases in all possible mission alternatives need to be computed as soon as possible to decide the next course of action.

Parameter	BDD size				Analysis time(s)			
Mission	Scenario 1		Scenario 2		Scenario 1		Scenario 2	
Scope	Phase 3	Mission	Phase 6	Mission	Phase 3	Mission	Phase 6	Mission
Scheme 1	2951	3170	3133	3174	4.9	5.3	5.3	5.4
Scheme 2	2741	2944	2822	2848	6.2	6.6	6.1	6.1
Scheme 3	2734	2807	2663	2691	5.0	5.2	4.4	4.4
Scheme 4	2541	2585	2501	2519	4.3	4.4	3.8	3.8
Scheme 5	2569	2620	2508	2527	4.3	4.4	3.6	3.6
Scheme 6	2690	2762	2743	2764	4.7	4.8	4.7	4.8
Scheme 7	4092	4590	4714	4747	13.2	14.6	14.2	14.3
Scheme 8	2936	3155	3118	3159	5.0	5.5	5.1	5.1
BOI	3514	3682	3567	3615	0.1	0.5	0.0	0.1

Table 6.4.3: Exact analysis for two mission alternatives

6.4.3 Summary

All the variable ordering schemes developed in Chapter 4 can be applied to PMS with multiple failure mode components (like the UAV in this case study). Scheme 1 to Scheme 8 are extensions of schemes applied to standard fault trees while the BOI scheme is a novel scheme.

When the current mission is too unreliable to continue and alternative mission configurations need to be considered, if there is enough time to calculate the conditional unreliabilities of future mission phases and the entire mission unreliability, Scheme 5 has the greatest potential to provide the fastest analysis, which can be seen from Table 6.3.1, where Scheme 5 leads to the least time to analyse the mission unreliability and the smallest BDD size and also from Table 6.4.3, where it leads to the smallest BDD size (and thus implies the least time to analyse the entire mission). If the decision is needed in a short

time, there may be only time allowed to analyse the conditional unreliability of the first altered phase in the mission alternative phase. In this case, the efficiency advantage of the BOI scheme is more convincing, since the online effort and time to re-order all variables and to construct BDDs for previous phases can be saved. This makes the BOI scheme the optimal choice when the immediate next course of action must be decided for the PMS based on exact conditional unreliabilities of the first altered phases in all possible alternative missions.

6.5 Choose An Optimal Mission Alternative From the Configuration Set: Approximation Analysis

6.5.1 Introduction

If during the performance of the SAR mission, a failure is detected in the fuel supply system of the UAV, this may have a serious impact on the phases of the current mission that are still to be performed. The influence of the detected failure on the mission unreliability can be great and any delay in making an instant decision as to the next course of action could lead to a catastrophe. Under this circumstance, approximation models are needed to achieve a faster mission unreliability analysis with reasonable accuracy loss so that a reasonable decision can be made as quickly as possible. To illustrate the advantage of the approximation analysis to help the UAV to choose the optimal alternative, four alternative missions are analysed after the detection of the fuel supply system failure using the approximation methods developed in Chapter. 5 and the exact BDD Model 3. If the original mission needs to be changed from phase 6, then the alternative mission configurations are shown in Table 6.5.1 to Table 6.5.4.

Phase	1	2	3	4	5	6	7	8	9	10
Fault Tree	T1	T2	T3	T4	T5	T6	T4	T5	T6	T7

Table 6.5.1: Configuration of alternative mission 1

Phase	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Fault Tree	T1	T2	T3	T4	T5	T6	T7	T4	T6	T3	T2	T1	T5	T4	T3	T2	T1

Table 6.5.2: Configuration of alternative mission 2

Phase	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Fault Tree	T1	T2	T3	T4	T5	T6	T5	T4	T6	T7	T6	T3	T2	T1

Table 6.5.3: Configuration of alternative mission 3

Phase	1	2	3	4	5	6	7	8	9	10	11	12
Fault Tree	T1	T2	T3	T4	T5	T6	T4	T7	T5	T3	T2	T1

Table 6.5.4: Configuration of alternative mission 4

6.5.2 Approximation Analysis

Since a decision is required urgently in this case, meaning the analysis speed is more important than the accuracy, a truncation limit of 10^{-1} is chosen for approximation Method 3, as discussed in Section 5.7, where 10^{-1} was seen to be the optimal truncation limit when considering I_λ when $\lambda = 0.4$, meaning that speed is more important than accuracy.

Table 6.5.5 shows the upper bound approximations to the mission unreliability computed using approximation Method 1 and Method 2, and the lower and upper bound approximations to the mission unreliability computed using approximation Method 3, along with their corresponding analysis time. It is seen from columns 2 to 5 that the time improvement is great when using any of the approximation models compared with the exact analysis. The accuracy of each approach varies: Method 1 should not be applied since it always produce an unreliability near to 1 regardless of how much time it could save; all the other methods can provide reasonable mission unreliability approximations compared with the exact unreliability result.

Parameter	Mission analysis time (s)				Mission unreliability			
Alternative	Mission 1	Mission 2	Mission 3	Mission 4	Mission 1	Mission 2	Mission 3	Mission 4
Method 1	0.1	0.1	0.1	0.1	$9.99 * 10^{-1}$	$9.99 * 10^{-01}$	$9.99 * 10^{-01}$	$9.99 * 10^{-1}$
Method 2	1.3	2.0	1.0	1.0	$3.26 * 10^{-06}$	$1.42 * 10^{-03}$	$3.18 * 10^{-04}$	$3.53 * 10^{-04}$
Method 3(Lower)	3.8	10.9	2.5	2.3	$2.10 * 10^{-06}$	$4.62 * 10^{-04}$	$1.81 * 10^{-04}$	$2.01 * 10^{-04}$
Method 3(Upper)	2.8	8.3	2.2	2.0	$3.08 * 10^{-06}$	$1.54 * 10^{-03}$	$2.31 * 10^{-04}$	$3.79 * 10^{-04}$
Exact	27.7	49.7	21.8	7.4	$2.47 * 10^{-06}$	$4.98 * 10^{-04}$	$1.94 * 10^{-04}$	$2.01 * 10^{-04}$

Table 6.5.5: Approximation analysis for four mission alternatives

The percentage of time reduction gained when using each approximation method compared with the exact mission analysis time is shown in Table 6.5.6 and illustrated in Figure 6.5.1. The accuracy of each approximation method compared to the exact mission analysis time is shown in Table 6.5.7 and illustrated in Figure 6.5.2.

For the four mission configurations analysed, all of the approximation methods save at least 68.96% of the analysis time reducing the time taken from minutes to seconds in each case. Regardless of the time savings achieved using Method 1, it is still not a practical

approximation method since the accuracy of the approximation is too low, i.e., always below 0.05%. For the other methods, the time improvements are good, while in terms of accuracy, method 3 (lower) always provides the closest approximation to the exact mission unreliability.

	Mission 1	Mission 2	Mission 3	Mission 4
Model 1	99.51%	99.82%	99.65%	99.00%
Model 2	95.36%	96.06%	95.50%	87.15%
Model 3(Lower)	86.39%	78.16%	88.36%	68.96%
Model 3(Upper)	89.76%	83.39%	89.87%	72.46%

Table 6.5.6: Analysis time improvement

	Mission 1	Mission 2	Mission 3	Mission 4
Model 1	0.00%	0.05%	0.02%	0.02%
Model 2	75.92%	35.17%	61.15%	57.10%
Model 3(Lower)	84.96%	92.62%	93.17%	99.91%
Model 3(Upper)	80.20%	32.36%	84.11%	53.11%

Table 6.5.7: Mission unreliability accuracy

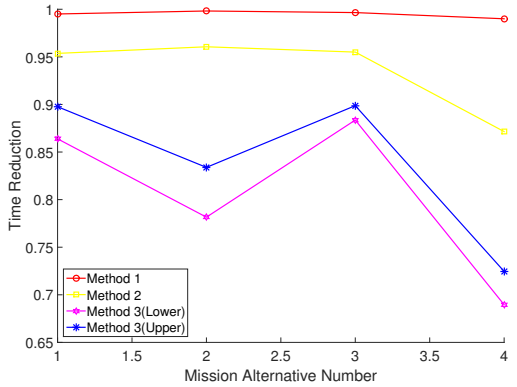

Figure 6.5.1: Comparison of time reduction

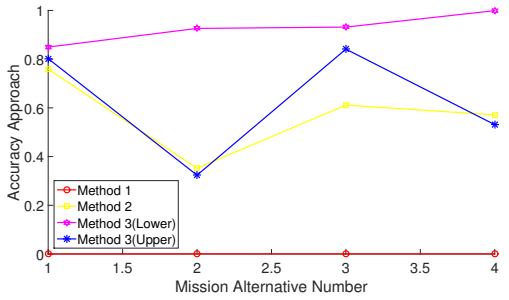

Figure 6.5.2: Comparison of accuracy approach

In order to choose an optimal mission alternative, all the 4 possible alternative mission configurations are analysed. Method 3 is suggested to be used as the reliability approximation method to support the UAV making a decision as to choose the optimal alternative mission, since it can provide both lower and upper bound approximations for the mission unreliability and also the conditional phase unreliability (when needed). Method 3 (Lower) provides the lowest accuracy loss when compared with the other approximation methods and much faster analysis than the exact mission unreliability analysis. Since the alternative mission 1 gives the smallest mission unreliability, it should be chosen for the UAV to continue.

6.5.3 Summary

When serious failures occur and urgent decisions are required about the future mission behaviour, approximation methods are needed to calculate the mission unreliability for all possible mission alternatives as fast as possible in order to select the one that will ensure the highest rate of mission success, i.e, the one with the smallest mission unreliability.

Of the developed approximation methods, Method 1 and Method 2 provide upper bound approximations whilst Method 3 provides both upper and lower bounds approximations of the entire mission unreliability. Method 1 can save large amounts of analysis time, however, the accuracy loss is so big that it makes this method impractical. The huge accuracy loss would cause unsound decision making and therefore could lead to big economic losses and casualties in the worst case. Both Method 2 and Method 3 can save a lot of time taken to analyse the unreliability of alternative missions when compared to the exact analysis models. This result has demonstrated a significant efficiency advantage of approximation Method 2 and Method 3 over the exact analysis method with reasonable accuracy loss. Method 3 (lower) provides the closest approximation values to the exact results. Due to all the advantages of Method 3, it would appear to be the most promising approximation method to be used as support tool in the decision making process.

6.6 Summary

The capability of the developed models and methods are demonstrated through application to the case study described in this chapter. The results have shown the improved analysis efficiency of the 3 developed BDD models, Model 1, Model 2 and Model 3 by comparing their analysis speed and constructed BDDs sizes with the existing Forward-BDD model. Nine different ordering schemes are compared by testing on the example SAR mission. BDD Model 3 and Scheme 5 are shown to offer the fastest analysis for the entire mission unreliability and thus would appear to be the most promising methods to be used as a decision support tool when exact analysis results are required. The efficiency advantage of the novel variable ordering scheme, BOI, is demonstrated by application in a scenario when it is required to analyse the exact unreliabilities of the first altered phases of all possible alternative missions in order to support the UAV in making a decision on the best next course of action. In the scenario when near-instant decisions are required, the three developed approximation methods are tested and compared with the exact analysis. Both Method 2 and Method 3 can provide practical approximations to mission unreliabilities, meaning faster analysis speed with reasonable accuracy loss when compared with the exact analysis. By adopting the models and methods developed during the course of this research, great efficiency improvements can be achieved when analysing updated mission unreliabilities, and thus it should be possible to make sound decisions based on the calculated unreliabilities more quickly than was possible in the past.

Chapter 7

Conclusions and Future Work

7.1 Summary and Conclusions

Increasing levels of autonomy will see many autonomous system need to improve their ability to interpret their environment and make decisions as to the best next course of action whenever failures or external threats are experienced. Reliability analysis can be used as an input to decision making processes by considering the systems to operate phased mission and carrying out phased mission analysis to calculate the success probability of the individual mission phases and the mission as a whole. Fast and accurate computation will form a crucial element in the decision making process. Therefore, the main objective of this research is to develop appropriate models to analyse the reliability of phased mission systems quickly and accurately, with a view to providing a reliability analysis methodology that could be used in a real-time decision support tool for systems operating phased missions in changing environments. The research goal is achieved by first reviewing risk and reliability definitions, traditional fault tree analysis, the BDD-based fault tree analysis for standard fault trees, and the use of a reliability analysis methodology as a decision making tool for PMSs. Then, different BDD models for phased mission systems are reviewed and further developed to improve the efficiency and accuracy of the analysis. Since the size of a BDD has a significant effect on the time required to quantify it and the BDD size is influenced by the variable ordering, nine different variable ordering schemes are investigated in this research. Eight of the schemes are extended from those applied to standard fault trees while one is newly developed. Due to the potential time limitation for decision making, approximation methods are investigated to allow the computation of the reliability of mission alternatives in a reasonable time frame. Three new approximation

models are developed in Chapter 5 and their analysis efficiency is compared to exact analysis by testing a large number of PMSs.

7.1.1 BDD Models for PMS with Multiple Failure Mode Components

For the non-repairable phased mission systems considered in this research, dependencies may exist due to mission phases and multiple component failure modes, meaning that BDD models considering such features need to be investigated. In Chapter 3, two existing BDD models are reviewed: the DEP-BDD model [43] and the Forward-BDD model [36]. In [36], the researchers developed the Forward-BDD model, which has been shown to provide improved efficiency and accuracy compared with DEP-BDD model developed in [43] in the analysis of PMS with multiple failure mode components. In order to perform the reliability analysis of such PMS as quickly and accurately as possible, previous models are reviewed and then further developed. The DEP-BDD rules are shown to lead to inaccurate quantification results [35][36]. In Chapter 3, two changes to the DEP-BDD model are proposed in order to ensure that accurate results can be obtained. One solution is to add a reduction process after constructing the DEP-BDDs and the other is to construct the DEP-BDDs using a different variable ordering which gives priority to dependencies within a component (multiple failure modes). For the Forward-BDD model, which gives the exact results, a new quantification method is proposed to speed up the analysis. Of the four BDD models which give exact analysis results, the two improved BDD models and the F-BDD model with newly-proposed quantification method are proved to always be faster than the Forward-BDD model and Method 3 has higher chance to lead to a faster reliability analysis than the other two new developed models and thus appears to be the most promising exact BDD model to be used as a decision support tool in PMS.

7.1.2 Variable Ordering Schemes for PMS

Since the size of a BDD has a significant impact on the time taken to quantify it, and the order of variables considered when constructing a BDD influences its size, variable ordering schemes for phased mission systems are investigated in order to minimise the BDD size and therefore attempt to speed up the reliability analysis in order to help the PMS making a faster and sound decision. In Chapter 4, eight different variable ordering schemes used during the construction of BDDs for standard fault trees are reviewed. All eight schemes are extended to allow variables to be ordered for phased mission fault trees.

Then, a new variable ordering scheme, BOI, which ensures a consistent variable ordering list no matter how the mission configuration changes, is proposed. Testing results on a large number of phased mission systems illustrate that the proposed ordering scheme leads to the fastest conditional unreliability analysis on the first altered phase of the alternative mission when current mission is no long safe to continue and possible alternative missions must be considered to help the PMS choose the optimal one to avoid any system failure.

7.1.3 Approximation Models for PMS

Due to the requirement for real-time analysis and the potentially limited amount of time available for decision making, approximation methods are needed in order to carry out reliability analysis of mission alternatives in a reasonable time frame. In chapter 5, three different methods are developed to approximate the unreliability of individual phases or the whole mission. Method 1 is developed based on the the early stage phased mission analysis method presented in [47], where mission unreliability is calculated using BDDs for individual phases without taking dependencies across phases into consideration. In Method 2, construction rules for Z-BDDs, BDDs which encodes only the minimal cut sets of the phased mission fault trees, are developed together with a corresponding quantification method. By quantifying the root node of the constructed Z-BDD, the rare event approximation to the mission unreliability can be obtained. Method 3 developed ran algorithm to construct truncated BDDs under certain truncation limits to reduce the size of the constructed BDD and therefore shorten the analysis time. A novel technique is also developed to choose an appropriate truncation limit analytically in order to control the loss of accuracy between the approximated lower bound approximation probability and the exact unreliability. Method 3 could be used to give both lower and upper bound approximations for mission unreliability and conditional phase unreliability. The testing results show that both Method 2 and Method 3 can greatly improve on the analysis time with reasonable accuracy loss compared with exact phased mission analysis.

7.1.4 Application of Investigated BDD Methods as A Decision Making Tool

In a search and rescue mission where a UAV is involved, fast and accurate reliability analysis of the future phases and the mission as a whole is a key factor in making decisions on the best next actions once internal failures or external threatens are detected. The BDD

models developed in Chapter 3 and the variable ordering schemes investigated in Chapter 4 are used to calculate the mission unreliability of the initial mission configuration before the SAR mission starts and BDD Model 3 and Scheme 5 are then chosen to perform updated mission unreliability when a mission is underway, since they appear to be the most promising to provide the least mission unreliability analysis time. The efficiency advantage of the developed BOI was demonstrated by a scenario when only time is allowed for calculating the exact unreliability of the first altered phases of all possible mission alternatives when new components are involved. Since variables ordering are done off-line and analysis of previous phases shared by the current and the alternative missions are already done, BOI scheme shows to improve analysis speed compared with the analysis using the other schemes. The scenario when the decision making time is very limited is also considered. The approximation models developed in Chapter 5 are applied to demonstrate the analysis speed advantage with reasonable loss of accuracy compared with exact analysis.

7.1.5 Conclusions

This research has reviewed a number of BDD models to analyse the reliability of PMS. Since the PMS considered in this research is non-repairable and contain multiple failure modes components, BDD models while taking these dependencies into account are investigated. Three models are developed and their performance is compared with the existing Forward-BDD model, which can provide exact analysis results for PMS containing multiple failure mode components. Model 3 is proved to offer the fastest mission unreliability analysis after testing all of the four models on a large set of sample PMS.

Eight variable ordering schemes applied to analyse standard fault trees are reviewed and then are extended to phased mission fault trees. The BOI is proposed with a view to provider a faster analysis of the the alternative missions in order to help the PMS making a decision on the best next course of action. The variable order list obtained by the BOI scheme remains the same no matter how the mission configuration changes, which will save the time of analysing previous phases shared by the current and the alternative missions. The BOI scheme has been proved to have improved analysis speed for unreliability analysis of the first altered phase in the mission alternative compared with the analysis with other schemes.

In cases where decision time is very restrict (near-instant), three approximation meth-

ods are developed in order to conduct a rapid reliability analysis with a trade off between reasonable loss of accuracy and the analysis speed. Method 2 and Method 3 have proved to be able to provide more efficient analysis than the exact mission analysis while with small loss of accuracy. Approximation method 3 is the most promising to be used as a support tool for PMS to make immediate decisions, since it can provide both lower and upper approximations to conditional phase unreliabilities and mission unreliability and has shown to be the most efficient when taking into account both the accuracy and the analysis speed.

The developed BDD models and techniques have shown significant improvement in analysis time through an application to a case study containing scenarios with different decision making requirements.

7.2 Future Work

Despite the contributions in the course of this research, there are several areas where future work can be carried out.

7.2.1 The Variable Ordering Schemes

Previous research has investigated the relation between the characteristics of fault trees and the performance of variable ordering schemes [9][7][8]. Though 9 variable ordering schemes are developed and tested on a large number of sample systems, further work could be done to investigate the relation between characteristics of the phased mission fault trees and the choice of the variable schemes. The variable ordering scheme can then be chosen according to the updated phased mission configuration characteristics rather than according to the average performance of the tested systems. The variable ordering has three levels: component level, phase level and failure mode level. The work done mainly focuses on the investigation of component level ordering, which is expected to have the majority impact on the converted BDDs sizes. Impact of variable ordering on BDDs sizes considering all the three levels still worth investigating in future.

7.2.2 Repairable Systems

In some systems, such as a land vehicle with maintenance staffs on board, some components can be repaired during the performance of a phased mission, meaning that a component that fails during a phase can return to a normal working state at a later time after repair.

In repairable systems, the BDD models developed in this research are no longer valid. The fault tree representation is no longer optimal since variables in the tree are no longer independent to each other. New models considering the repairable features of the system need to be investigated, such as petri-net models or combination models of petri-net and BDDs.

7.2.3 Application to Real Systems

Due to the lack of access to real systems, the developed models are tested on randomly-generated example PMS using the algorithm. The developed methods could be applied to real PMS in order to further test the validity of the proposed approaches and demonstrate the application of the developed techniques.

7.2.4 Multiple Platform PMSs Analysis

In scenarios such as a network-centric warfare, several platforms are required to cooperate in order to achieve a common objective. Whist some literature reviews and simple analysis techniques are demonstrated in this project, further investigation is required for PMSs with multiple collaborating platforms. The analysis may be extended to consider the communication between platforms, optimal arrangement of platform and phases within the mission and the alteration of mission configurations.

Appendix A

Algorithm for Random Fault Tree Generating

This section introduces the algorithms used to generate random benchmark fault trees used for tests throughout this thesis. This method was firstly presented in [24], and summarised in here for completeness of this thesis.

The main method is presented in Algorithm A.1 , and some details are listed as follows:

1. Determine gate's logic:

   the fault tree consists of a series of OR gates and AND gates. The logic structure of the fault tree is closely related to the top gate's logic (presented by parameter *topLogic*).

   If *topLogic* is OR, then

   Gates in 'odd' layers are all OR gates,

   Gates in 'even' layers are all AND gates.

   If *topLogic* is AND, then

   Gates in 'odd' layers are all AND gates,

   Gates in 'even' layers are all OR gates.

2. Determine gate's fanout number:

   All the gates' fanout number follow a uniform distribution: $U[n_{min}, n_{max}]$.

3. Determine node's type:

Denote the probability of a node in layer i being gate with P_i^{gate}, then any node's type is determined as follows:

First, draw r from the uniform distribution over $[0, 1]$.

If $r < P_i^{gate}$, the node is a gate. Otherwise it is an event.

4. Determine event's type and failure mode:

The total number of distinct events in the tree is denoted as N and the type of each event is determined by the probability of being multiple failure mode event P^{mul} as follows:

Before the algorithm starts, for each distinct event $j = 1, 2, ..., N$, draw r from the uniform distribution $U[0, 1]$ and label j as event with multiple failure modes if $r < P^{mul}$.

During the algorithm, select the event uniformly from $1, 2, ..., N$ and check if it is a multiple failure mode event.

For each multiple failure mode event, label its failure mode as m, where m follows the uniform distribution $U[1, m_{max}]$ and m_{max} is the maximal number of failure modes.

Algorithm A.1 Random fault tree generator

Input: *topLogic*, P_i^{gate}, n_{min}, n_{max}, P^{mul}, N, m_{max}.

```
Generate top gate root
Determine root fanout number and logic
Push(root)
if stack is empty then
    Return root.
end if
Pop(g)
Determine g's fanout number and logic
for each g's fanout do
    Generate node x
    Determine x's type (gate or event)
    if x is a gate then
        Push(x)
    else if x is an event then
        Determine x's failure mode type (single or multiple)
        If x is multiple, generate x's failure mode
    end if
end for
Go to step 4.
```

Appendix B

Testing results

B.1 Testing results for phased mission models

	BDD sizes			Analysis times			
Mission	Model 1	Model 2	Model 3 and Forward-BDD	Model 1	Model 2	Model 3	Model 4
1	4593	4593	4068	3.41	3.50	1.27	99.80
2	1598	1585	1731	0.42	0.41	0.55	20.19
3	2448	2384	2278	1.07	0.98	1.22	17.41
4	2910	2714	2543	1.70	1.33	1.57	18.11
5	2302	2164	2136	1.01	0.76	0.99	17.14
6	2622	2426	2311	1.36	0.99	1.29	17.94
7	2202	2064	2044	0.93	0.70	0.92	18.15
8	1917	1917	1212	1.21	1.29	0.53	3.32
9	2971	2971	1754	4.26	4.26	0.96	5.75
10	540	542	527	0.02	0.02	0.02	0.03
11	1924	1896	1864	0.34	0.98	0.44	4.87
12	6876	5381	5498	4.87	10.51	5.22	82.32
13	11422	8648	8448	12.68	14.50	8.62	212.88
14	5385	4382	4083	3.26	5.79	2.45	38.68
15	1148	1150	1107	0.15	0.14	0.16	0.96
16	6156	6175	5490	0.34	0.38	0.27	2.14
17	14548	33724	15418	9.75	123.61	12.75	388.28
18	24244	97984	24554	31.86	1032.53	35.26	1377.33
19	22647	87047	26535	28.24	840.02	40.53	1726.56
20	15530	39149	19198	11.19	185.34	23.80	792.32
21	7025	7414	5900	2.29	3.12	0.92	11.69
22	623	623	546	0.03	0.02	0.02	0.05
23	6787	6731	4758	8.58	9.88	4.06	85.39
24	21539	69348	28608	96.13	1036.51	144.30	5331.12
25	2492	2393	2368	2.67	2.73	1.14	17.20
26	3230	3230	3230	0.20	0.20	0.20	0.59
27	8728	8728	10350	7.15	7.15	16.48	222.86
28	12196	12196	16322	11.88	12.20	31.35	503.28
29	8714	8714	10975	6.94	6.97	16.87	260.60
30	4227	4227	4432	0.65	0.65	0.72	5.73
31	1797	1799	1706	0.05	0.05	0.05	0.06
32	3678	3912	3694	0.33	0.72	0.47	2.48
33	5529	9455	6224	1.24	7.04	2.11	19.27
34	8809	29447	12874	3.17	64.60	9.47	134.87
35	4171	4773	4787	0.62	1.51	1.10	7.97
36	2070	2089	1962	0.11	0.13	0.11	0.47
37	1194	1194	1206	0.03	0.03	0.02	0.08

38	20909	22961	24449	38.30	92.49	78.41	3403.49
39	692	688	768	0.03	0.03	0.04	0.21
40	7456	6491	7250	12.13	7.11	11.66	234.68
41	27086	18838	20129	207.69	65.72	89.30	2244.75
42	9098	8409	10136	18.27	10.72	19.55	411.98
43	9521	7999	9721	14.12	8.44	16.99	391.27
44	3305	3230	2991	1.33	1.22	1.00	14.22
45	584	574	465	0.02	0.02	0.01	0.04
46	6541	7206	4118	4.79	10.39	1.88	24.51
47	30365	31380	11782	97.32	238.58	19.33	385.39
48	31192	169552	50991	66.86	7825.87	289.63	8135.84
49	15152	14791	5489	24.01	40.92	3.29	54.73
50	2572	2514	2308	0.94	1.00	0.44	3.67
51	599	600	618	0.02	0.02	0.02	0.05
52	3362	3449	3701	0.86	1.12	1.42	17.06
53	7053	9767	10095	5.60	15.25	14.89	296.28
54	20226	27372	29843	41.91	116.14	124.73	3411.45
55	4184	4401	5048	1.26	1.97	2.69	33.32
56	2377	2372	2236	0.27	0.27	0.25	1.50
57	1506	1498	1479	0.13	0.25	0.22	3.34
58	2377	2372	2236	0.27	0.27	0.25	1.48
59	732	584	571	0.07	0.03	0.03	0.15
60	6369	4551	5708	13.83	2.44	3.26	48.31
61	15363	24428	21748	16.34	49.23	41.08	726.75
62	9897	7007	8182	7.92	6.50	6.57	113.38
63	2418	2235	2070	0.58	0.42	0.39	3.47
64	554	552	523	0.03	0.03	0.02	0.08
65	5511	5330	5778	6.57	6.61	7.60	140.01
66	12168	9745	11306	33.47	22.42	26.33	480.72
67	9559	8781	9047	20.90	20.15	9.96	165.39
68	719	709	765	0.09	0.09	0.08	0.41
69	5870	4325	4954	5.62	4.10	4.25	75.44
70	26669	13839	14911	255.05	89.77	38.75	919.23
71	17900	6746	6990	89.52	14.53	7.71	150.43
72	2108	1901	2131	0.75	0.42	0.48	5.37
73	566	571	456	0.03	0.03	0.02	0.05
74	3166	3082	2200	2.78	2.48	0.59	4.77
75	8559	7080	3955	17.17	15.08	2.11	22.48
76	22385	17988	6900	303.24	89.18	5.31	66.50
77	4954	4535	2770	5.49	4.15	1.02	9.67
78	1606	1632	1385	0.16	0.16	0.08	0.43
79	606	625	560	0.03	0.02	0.00	0.02
80	4285	5226	3470	1.25	2.05	0.69	4.72
81	24381	49858	8439	40.07	168.55	4.31	54.22
82	44330	240660	14741	135.89	3547.62	13.02	172.77
83	5089	7911	4857	1.34	5.28	1.46	14.11
84	2223	2423	1970	0.34	0.55	0.17	1.13
85	612	615	598	0.08	0.08	0.07	5.04
86	2193	2131	1745	0.95	0.84	0.40	23.51
87	3373	3124	2381	2.24	1.87	0.74	26.34
88	5624	5331	3414	6.64	6.04	1.52	28.80
89	2374	2295	1872	1.06	0.94	0.45	26.11
90	1467	1457	1335	0.36	0.35	0.24	22.26
91	753	759	743	0.19	0.20	0.20	60.39
92	3831	2985	2608	5.08	3.78	1.76	537.50
93	7014	4736	3339	15.16	7.52	3.69	546.40
94	26248	8448	4473	253.73	29.84	7.55	555.27
95	4296	3230	2715	6.09	4.46	2.04	543.79
96	2146	2159	2138	1.04	1.07	1.13	462.23
97	3863	3755	4103	3.94	3.71	4.86	1462.03
98	11863	9981	7167	30.48	17.19	7.86	744.28

99	5760	5167	3287	7.62	7.39	2.50	120.95
100	3790	3572	2461	3.30	3.98	1.22	119.14

Table B.1.1: Sizes of mission failure BDDs and analysis time for the whole phased mission using the 4 BDD models for each of the tested phased mission

B.2 Testing results for ordering schemes

Mission	Scheme 1	Scheme 2	Scheme 3	Scheme 4	Scheme 5	Scheme 6	Scheme 7	Scheme 8	BOI
1	539	603	541	541	484	542	446	471	939
2	4471	3554	2523	3397	2285	3627	5804	3072	9727
3	18586	17204	11617	12424	10871	20226	17204	14336	47711
4	14099	11923	6744	10493	6943	11240	32807	8864	49750
5	10617	8093	4867	9262	4028	6757	18245	7723	34701
6	2210	2261	1748	1592	1516	2129	1928	1612	3188
7	5380	5771	5711	5711	5103	4870	3992	4122	6124
8	10554	11072	9640	9903	8419	7414	10446	8346	12121
9	11460	13172	11086	11493	10526	10799	13172	9912	13992
10	11677	13169	11083	11490	11065	11820	13169	9909	13989
11	10663	11124	9826	10072	9273	10546	11124	8381	12388
12	5651	6737	5872	5915	5323	5079	7567	5083	6879
13	553	586	527	527	482	568	473	481	740
14	4986	9698	6134	3775	4666	4472	4659	3894	16456
15	7989	30487	14906	7440	8402	8395	30487	6920	42057
16	4147	9585	5963	3563	3700	4179	4105	3352	14151
17	3555	3571	3375	3380	3706	3687	2987	3282	4361
18	8560	7186	6690	6898	8513	7393	6685	7226	8368
19	9187	7931	7401	7875	9434	9158	7931	8041	9822
20	8673	7260	6703	6925	7902	8093	7260	7312	8689
21	4839	4344	4032	4121	4938	4781	4981	4278	4866
22	2007	1942	1765	1765	1917	1945	1965	1966	2255
23	3573	3409	3293	3547	3512	3482	3326	3229	3690
24	4366	4173	3880	4185	4278	4132	4173	3884	4619
25	5878	5728	5371	5836	5918	5714	5728	5427	6478
26	3453	3463	3427	3611	3663	3670	3463	3307	3729
27	2300	2216	1965	2148	2198	2107	1990	1986	2483
28	1414	2471	2339	1234	1485	1306	1524	1310	1569
29	8561	28874	23629	12550	7635	9749	23633	8076	32571
30	658	566	625	770	608	757	746	578	702
31	2536	3455	2702	3366	2106	2734	2702	2114	3679
32	3578	3682	3575	4378	3113	4037	3682	2883	5177
33	2958	3641	2919	3613	2438	2947	3641	2338	4460
34	2888	3597	2871	3561	2339	3308	3597	2273	3877
35	1835	2072	1913	2232	1578	2252	1990	1478	2341
36	602	582	581	501	613	501	555	449	430
37	2846	2898	2724	2759	2575	2335	2724	2558	3035
38	4048	4257	3969	4238	3811	3833	4257	3748	5164
39	5275	5639	5118	5811	5883	5216	5639	4895	6783
40	2996	3215	3112	3022	2831	2874	3215	2873	3356
41	2146	2160	1999	2225	2027	1923	2169	2074	2671
42	716	715	717	661	651	653	605	628	745
43	2995	2922	2926	3014	2983	3137	2926	3246	3175
44	4150	4381	4469	4576	4560	4538	4381	4841	4400
45	5348	5536	5704	5857	5491	5741	5536	6282	5409
46	3525	3412	3497	3436	3634	3721	3412	3774	3630
47	2190	2202	2281	2355	2297	2397	2565	2446	2094
48	1359	1262	1282	1742	1410	1268	1271	1629	1420
49	7255	7219	6480	7100	4776	5091	6755	5870	6905
50	544	595	527	588	481	614	542	502	813

51	2141	3535	3577	3045	2094	2227	3577	2069	7069
52	2992	6019	6380	4330	2870	3808	6019	3225	9559
53	4151	7892	8518	5606	3712	5145	7892	4159	11159
54	2799	3832	3870	3301	2200	2502	3832	2333	7906
55	1684	2730	2833	2344	1446	1928	1469	1652	3758
56	588	607	553	516	550	552	552	547	725
57	2242	2265	2268	3000	1918	2199	2268	2121	3345
58	2882	3213	3443	3642	2916	2784	3213	2905	4476
59	3620	4033	4470	4191	3263	3773	4033	3646	5326
60	2193	2651	2903	3308	2232	2541	2651	2485	3957
61	1538	1738	1708	2041	1777	1497	1850	1662	2110
62	672	702	710	846	731	712	829	801	778
63	2690	2312	2155	3054	2033	1895	2155	2260	3830
64	3520	3068	2951	4022	2904	3028	3068	3066	5802
65	4180	3746	3624	4776	3671	4614	3746	3776	6880
66	2941	2600	2482	3434	2514	2481	2600	2537	5188
67	1902	1745	1662	2172	1571	1662	1817	1710	2687
68	472	376	386	471	529	481	469	472	599
69	2051	1750	1753	1917	1736	1800	1753	1853	1905
70	2972	2756	2762	2638	2546	2563	2756	2866	2650
71	3663	3657	3653	3329	3271	3387	3657	3850	3243
72	2276	2003	2006	2133	1962	1991	2003	2070	2196
73	1557	1403	1448	1414	1300	1339	1527	1461	1513
74	574	521	515	575	518	545	529	522	762
75	2704	2559	2483	2596	2026	2294	2483	2181	3641
76	3998	3765	3649	3944	2927	3226	3765	3382	6321
77	5215	4737	4602	5260	4084	4184	4737	4522	8273
78	3079	3049	2959	2935	2218	2564	3049	2487	4451
79	2081	2073	2006	2032	1520	1751	1806	1738	2955
80	687	611	611	639	666	569	548	636	747
81	1841	1476	1476	1688	1870	1855	1632	1876	1706
82	2621	2183	2183	2381	2451	2459	2183	2591	2291
83	3737	3142	3142	3396	3494	3288	3142	3644	3245
84	1914	1581	1581	1793	2110	1888	1561	1874	1815
85	1603	1202	1202	1341	1433	1424	1076	1556	1226
86	1104	944	944	1006	997	948	555	761	1085
87	2917	2401	2401	2622	2540	2831	2631	1961	2528
88	3702	3129	3129	3339	3164	3225	3129	3157	3313
89	4922	4261	4261	4473	4196	4087	4261	4233	4575
90	3034	2499	2499	2719	2795	3208	1693	2233	2634
91	2543	1974	1974	2143	2129	2057	1461	1826	2031
92	866	821	821	811	906	962	729	877	949
93	2789	2474	2474	2175	2985	2543	2670	2044	2374
94	4189	3920	3920	3287	3422	2926	3920	3532	3298
95	2876	2587	2587	2288	2863	2958	1846	2495	2480
96	5099	5204	5204	5061	4968	5069	4954	5085	5465
97	19024	18981	18981	18376	17517	18430	18379	18313	20216
98	33894	34223	34223	32925	30833	32816	34223	32619	36190
99	44308	44984	44984	43387	39980	41303	44984	43589	48341
100	25961	26023	26023	25109	23765	25940	26023	24606	26939

Table B.2.1: Mission failure BDD sizes for the nine different ordering schemes

Mission	scheme 1	scheme 2	scheme 3	scheme 4	scheme 5	scheme 6	scheme 7	scheme 8	BOI
1	0.0	0.1	0.0	0.1	0.0	0.2	0.1	0.4	0.2
2	6.2	3.5	1.4	4.3	1.1	3.9	11.4	5.3	31.4
3	112.4	98.2	42.4	53.3	45.0	168.0	113.3	68.2	749.8
4	68.0	56.7	14.4	40.3	15.2	44.2	403.6	27.9	844.9
5	37.8	25.3	7.0	29.0	4.9	15.5	120.6	21.2	405.9
6	1.1	1.1	0.5	0.7	0.5	1.3	1.0	3.1	2.5
7	0.9	1.6	1.7	1.9	0.7	1.6	1.6	12.3	2.2

8	18.0	14.8	9.1	10.5	4.2	5.0	12.3	21.9	21.1
9	18.4	29.8	14.7	18.0	8.1	26.9	31.6	24.8	51.9
10	24.6	25.9	14.6	17.8	8.0	17.5	27.9	23.7	56.9
11	18.5	18.1	11.3	11.2	5.6	16.7	19.4	23.4	37.5
12	2.5	4.6	3.1	3.6	1.6	3.7	7.1	13.8	5.0
13	0.1	0.1	0.0	0.1	0.1	0.2	0.1	0.7	0.1
14	7.2	27.1	9.0	4.7	5.3	5.8	10.5	11.2	104.1
15	31.1	424.5	83.8	22.1	27.6	29.4	432.7	25.1	818.2
16	4.3	24.0	8.0	2.9	3.2	4.5	4.0	8.1	59.5
17	1.0	0.9	0.7	1.0	0.9	2.2	1.5	5.8	1.5
18	15.1	8.1	6.8	9.6	10.9	11.0	7.5	17.3	11.2
19	20.7	16.7	11.0	12.7	15.3	21.5	18.1	21.8	22.5
20	20.1	12.7	8.5	10.1	7.6	14.6	13.8	19.2	17.0
21	3.5	2.8	2.1	2.9	3.0	4.4	5.8	10.4	2.8
22	0.2	0.3	0.2	0.6	0.2	0.6	0.8	2.7	0.4
23	1.1	1.1	1.2	2.5	1.2	1.5	2.0	4.0	1.5
24	2.1	2.6	2.4	3.9	2.2	3.0	3.7	5.7	3.6
25	4.7	4.9	4.6	7.1	3.6	5.1	6.8	6.8	8.1
26	1.5	1.7	1.9	2.9	1.6	2.1	2.7	4.6	2.3
27	0.5	0.6	0.5	0.9	0.5	0.8	1.1	3.7	0.6
28	0.2	0.4	0.4	0.2	0.2	0.5	0.4	2.2	0.3
29	23.4	329.2	216.4	77.5	10.6	21.8	215.4	70.7	462.5
30	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.5	0.1
31	1.5	3.8	1.7	3.3	1.1	2.7	2.5	3.2	3.5
32	3.8	5.8	3.4	5.8	2.4	6.5	6.4	4.6	10.7
33	3.2	7.5	2.4	3.8	1.2	3.5	8.3	4.1	7.8
34	2.6	6.0	2.1	3.8	1.1	4.2	6.7	3.7	5.6
35	0.6	1.1	0.7	1.2	0.4	1.5	1.2	2.6	1.3
36	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.2	0.0
37	1.8	1.4	1.2	2.2	1.2	2.2	1.8	3.9	2.7
38	5.8	3.1	2.7	5.7	3.1	4.3	4.0	7.0	9.7
39	11.4	6.4	5.3	11.4	10.2	8.9	7.9	10.3	19.8
40	2.3	1.6	1.5	2.9	1.4	2.3	2.2	4.6	4.1
41	0.8	0.6	0.6	1.2	0.6	1.0	1.4	3.1	1.2
42	0.1	0.0	0.1	0.1	0.1	0.2	0.1	0.5	0.1
43	1.1	1.1	0.9	2.2	0.8	1.7	1.5	4.1	2.3
44	2.9	2.9	2.6	5.0	2.6	4.2	3.8	7.5	6.0
45	5.9	5.6	4.7	8.8	4.5	7.0	6.6	11.2	10.8
46	1.5	1.7	1.6	2.8	1.4	2.4	2.3	5.0	3.6
47	0.5	0.5	0.5	1.1	0.5	1.1	1.6	3.4	0.7
48	0.4	0.3	0.4	0.7	0.5	0.6	0.6	0.9	0.4
49	13.7	13.8	10.8	15.4	5.2	6.8	12.8	14.3	12.7
50	0.0	0.1	0.1	0.1	0.0	0.2	0.2	0.5	0.2
51	0.9	4.0	4.1	3.1	0.9	1.7	4.7	3.0	19.1
52	2.8	13.9	15.5	6.6	2.0	5.4	15.5	5.8	43.8
53	5.8	32.9	38.3	12.3	4.1	11.1	34.4	9.2	78.9
54	2.3	6.2	6.2	4.0	1.2	2.4	6.7	3.9	29.0
55	0.5	1.9	2.1	1.4	0.4	1.2	0.8	2.5	4.1
56	0.1	0.1	0.1	0.1	0.1	0.2	0.2	0.4	0.1
57	1.1	1.0	1.1	3.7	0.5	1.5	1.7	2.5	3.6
58	2.0	3.5	2.7	5.2	2.2	3.2	4.2	3.2	8.8
59	2.9	6.9	5.8	7.3	1.9	5.1	7.8	3.9	16.2
60	1.1	2.0	1.9	4.6	1.0	2.4	2.7	2.9	6.1
61	0.5	0.6	0.6	1.4	0.8	0.9	1.4	1.7	1.1
62	0.1	0.1	0.1	0.2	0.1	0.2	0.2	1.0	0.1
63	2.1	0.9	1.0	3.4	0.8	1.3	1.5	4.2	4.4
64	4.5	2.7	2.3	6.5	2.4	3.4	3.5	5.8	13.3
65	7.3	4.8	3.9	10.7	4.7	10.3	6.1	7.9	26.1
66	2.8	1.5	1.3	4.2	1.3	1.8	2.2	4.4	8.9
67	0.8	0.6	0.5	1.3	0.5	0.9	1.0	3.0	1.7
68	0.0	0.1	0.0	0.1	0.1	0.2	0.1	0.6	0.1

69	0.9	0.4	0.4	1.0	0.4	0.9	0.9	3.0	0.9
70	2.0	0.9	0.9	1.7	0.8	1.6	1.6	3.7	1.9
71	2.7	1.7	1.6	2.6	1.1	2.6	2.7	4.1	2.8
72	1.3	0.5	0.6	1.2	0.4	1.1	1.1	3.4	1.4
73	0.3	0.2	0.3	0.5	0.2	0.5	0.8	2.4	0.4
74	0.0	0.0	0.0	0.1	0.0	0.2	0.1	0.4	0.1
75	1.1	1.8	1.6	1.2	0.5	1.2	2.2	2.0	3.3
76	3.0	4.8	4.4	2.5	1.0	2.7	5.6	3.1	11.2
77	5.1	9.2	8.4	4.2	1.8	4.3	10.2	4.4	23.7
78	1.5	2.8	2.5	1.6	0.5	1.7	3.4	2.2	5.1
79	0.5	0.9	0.9	0.6	0.3	0.8	0.8	1.8	1.6
80	0.1	0.1	0.1	0.1	0.1	0.2	0.1	0.8	0.1
81	0.8	0.3	0.3	0.5	0.7	1.2	0.8	1.6	0.6
82	3.2	0.9	0.9	1.2	1.3	2.1	1.1	2.8	1.2
83	8.8	1.7	1.7	1.9	2.7	4.0	2.0	4.4	2.5
84	0.9	0.4	0.4	0.6	1.1	1.2	0.9	1.5	0.8
85	0.7	0.2	0.2	0.4	0.4	0.8	0.5	1.3	0.3
86	0.3	0.2	0.2	0.3	0.3	0.6	0.1	2.1	0.3
87	2.4	1.4	1.5	2.2	2.0	4.3	2.7	4.7	1.9
88	5.7	3.8	3.6	4.4	4.1	6.9	3.6	7.6	4.8
89	11.5	6.5	6.7	8.2	7.4	14.0	7.0	9.9	9.0
90	3.8	2.1	2.1	2.4	2.6	6.5	1.1	6.0	2.4
91	1.6	1.0	1.1	1.4	1.2	2.3	0.5	4.1	1.0
92	0.2	0.1	0.2	0.2	0.2	0.6	0.2	1.3	0.2
93	1.9	1.1	1.1	0.9	2.5	2.6	2.6	4.1	1.4
94	8.9	5.5	5.7	2.9	4.6	5.2	6.0	8.5	3.2
95	2.9	1.8	1.6	1.1	2.4	3.5	1.2	5.7	1.8
96	0.8	0.8	0.8	1.1	0.8	3.8	1.4	10.2	0.8
97	7.9	7.5	7.9	9.1	7.7	31.7	11.4	55.1	8.3
98	20.7	20.0	19.9	23.0	19.9	67.8	26.9	64.9	21.8
99	34.9	34.0	33.9	40.5	33.0	88.2	42.9	80.6	40.9
100	13.5	12.6	12.7	15.4	13.0	53.0	18.0	61.1	13.9

Table B.2.2: Mission failure probability analysis time (in seconds) for the nine different ordering schemes

Mission	scheme 1	scheme 2	scheme 3	scheme 4	scheme 5	scheme 6	scheme 7	scheme 8	BOI
1	0.02	0.03	0.02	0.11	0.03	0.14	0.09	0.42	0.03
2	0.09	0.15	0.12	0.32	0.11	0.45	0.49	2.81	0.06
3	0.18	0.25	0.21	0.50	0.20	0.81	0.87	3.73	0.08
4	0.19	0.25	0.23	0.48	0.16	0.84	0.85	3.22	0.10
5	0.13	0.24	0.19	0.41	0.15	0.74	0.78	2.73	0.08
6	0.09	0.13	0.11	0.24	0.08	0.37	0.45	2.72	0.05
7	0.82	1.23	1.06	1.40	0.62	1.37	1.44	12.25	1.16
8	2.01	1.94	1.68	2.69	1.25	2.01	2.83	18.10	1.97
9	2.62	2.21	1.74	2.99	1.65	2.99	3.55	17.43	2.72
10	2.45	2.18	1.77	2.90	1.89	3.46	3.54	17.71	2.31
11	1.81	1.92	1.60	2.53	1.42	2.86	3.10	18.80	2.06
12	1.30	1.11	0.89	1.40	0.81	1.55	1.87	13.01	1.16
13	0.04	0.06	0.03	0.07	0.04	0.23	0.11	0.66	0.01
14	0.22	0.55	0.33	0.40	0.28	0.84	0.81	7.03	0.05
15	0.39	1.11	0.63	0.81	0.52	1.48	1.65	8.39	0.09
16	0.20	0.54	0.32	0.37	0.25	0.71	0.60	5.89	0.05
17	0.70	0.61	0.42	0.78	0.82	1.83	1.32	5.71	0.50
18	2.38	1.08	0.99	2.26	2.52	3.33	2.01	12.20	0.82
19	2.08	1.14	1.08	2.62	2.89	4.77	2.64	14.42	0.97
20	2.45	1.11	0.99	2.31	1.75	3.70	2.40	13.50	0.83
21	1.08	0.59	0.55	1.28	1.50	2.39	2.10	9.31	0.50
22	0.16	0.16	0.14	0.51	0.19	0.52	0.73	2.70	0.02
23	0.23	0.28	0.24	1.22	0.39	0.88	1.20	3.39	0.04

24	0.28	0.32	0.29	1.27	0.46	1.22	1.43	4.32	0.05
25	0.40	0.46	0.42	1.90	0.61	1.66	2.00	4.07	0.11
26	0.23	0.30	0.28	1.13	0.46	0.99	1.25	3.66	0.04
27	0.18	0.16	0.18	0.57	0.22	0.52	0.83	3.52	0.02
28	0.09	0.23	0.21	0.15	0.11	0.43	0.23	2.12	0.06
29	0.69	1.07	1.01	1.54	0.84	2.50	1.67	58.84	0.32
30	0.06	0.05	0.05	0.12	0.05	0.22	0.19	0.51	0.02
31	0.18	0.17	0.15	0.66	0.15	0.82	0.72	2.57	0.05
32	0.30	0.25	0.22	0.86	0.22	1.33	1.17	3.21	0.07
33	0.19	0.17	0.16	0.66	0.18	1.08	0.90	3.08	0.05
34	0.20	0.17	0.16	0.65	0.17	1.07	0.82	2.98	0.06
35	0.14	0.12	0.12	0.40	0.12	0.66	0.66	2.43	0.03
36	0.03	0.05	0.05	0.11	0.05	0.18	0.16	0.21	0.01
37	0.17	0.21	0.21	0.56	0.19	0.81	0.88	2.86	0.03
38	0.25	0.29	0.31	0.89	0.32	1.26	1.12	3.54	0.04
39	0.34	0.40	0.43	1.38	0.52	1.62	1.56	3.62	0.06
40	0.18	0.20	0.23	0.67	0.21	0.88	0.86	3.20	0.06
41	0.13	0.15	0.23	0.41	0.16	0.60	0.67	2.51	0.02
42	0.04	0.04	0.04	0.10	0.05	0.18	0.13	0.46	0.02
43	0.18	0.16	0.17	0.62	0.21	0.73	0.76	3.09	0.02
44	0.25	0.27	0.28	0.89	0.35	1.32	1.08	3.62	0.03
45	0.33	0.34	0.37	1.30	0.44	1.73	1.41	3.90	0.04
46	0.22	0.19	0.20	0.66	0.26	0.93	0.84	3.18	0.03
47	0.13	0.14	0.15	0.42	0.17	0.60	0.62	2.81	0.03
48	0.11	0.12	0.16	0.38	0.11	0.28	0.19	0.80	0.12
49	0.62	0.49	0.44	1.12	0.49	0.83	0.83	5.60	0.32
50	0.03	0.05	0.04	0.11	0.03	0.19	0.14	0.50	0.09
51	0.08	0.17	0.13	0.58	0.12	0.69	0.73	2.18	0.27
52	0.15	0.26	0.20	0.82	0.17	1.16	1.09	2.94	0.35
53	0.19	0.34	0.31	1.33	0.24	1.54	1.44	2.86	0.45
54	0.12	0.19	0.15	0.65	0.15	0.93	0.86	2.56	0.27
55	0.08	0.13	0.10	0.37	0.09	0.54	0.55	2.14	0.22
56	0.05	0.05	0.04	0.10	0.04	0.19	0.16	0.39	0.04
57	0.18	0.16	0.14	0.54	0.15	0.72	0.70	1.40	0.12
58	0.25	0.24	0.17	0.78	0.26	1.27	1.06	1.62	0.11
59	0.31	0.32	0.24	1.19	0.31	1.68	1.37	1.57	0.14
60	0.21	0.18	0.15	0.58	0.18	0.95	0.82	1.54	0.09
61	0.15	0.12	0.14	0.36	0.18	0.58	0.63	1.25	0.12
62	0.04	0.06	0.07	0.16	0.08	0.16	0.17	0.89	0.04
63	0.17	0.15	0.17	0.68	0.19	0.58	0.74	3.18	0.11
64	0.18	0.23	0.25	0.90	0.24	1.00	1.05	3.75	0.15
65	0.22	0.29	0.32	1.29	0.37	1.40	1.57	4.14	0.19
66	0.15	0.17	0.19	0.68	0.20	0.75	0.85	3.32	0.12
67	0.09	0.11	0.12	0.43	0.13	0.46	0.59	2.65	0.08
68	0.02	0.03	0.03	0.10	0.05	0.16	0.12	0.58	0.01
69	0.11	0.12	0.12	0.56	0.16	0.57	0.64	2.81	0.02
70	0.14	0.22	0.18	0.76	0.26	0.89	0.96	3.07	0.03
71	0.20	0.29	0.24	1.16	0.29	1.27	1.26	3.02	0.04
72	0.14	0.13	0.11	0.56	0.16	0.71	0.75	3.06	0.05
73	0.08	0.10	0.08	0.34	0.10	0.41	0.58	2.32	0.02
74	0.03	0.03	0.03	0.11	0.03	0.15	0.13	0.40	0.02
75	0.13	0.11	0.11	0.54	0.16	0.60	0.72	1.66	0.07
76	0.23	0.17	0.17	0.84	0.23	1.08	0.97	1.91	0.12
77	0.31	0.23	0.23	1.25	0.38	1.28	1.27	2.00	0.13
78	0.15	0.13	0.13	0.61	0.16	0.82	0.76	1.74	0.08
79	0.12	0.08	0.08	0.38	0.15	0.47	0.61	1.64	0.05
80	0.05	0.05	0.05	0.06	0.06	0.23	0.08	0.76	0.03
81	0.12	0.09	0.10	0.26	0.27	0.66	0.30	1.07	0.08
82	0.35	0.18	0.23	0.43	0.41	1.19	0.41	1.49	0.08
83	0.43	0.25	0.26	0.63	0.48	1.47	0.53	1.59	0.11
84	0.13	0.10	0.10	0.24	0.24	0.71	0.37	1.08	0.06

85	0.17	0.09	0.10	0.20	0.18	0.57	0.23	1.02	0.04
86	0.14	0.10	0.12	0.13	0.18	0.56	0.09	2.05	0.20
87	0.28	0.19	0.20	0.35	0.31	1.60	0.39	4.19	0.25
88	0.36	0.27	0.28	0.41	0.36	2.53	0.49	5.62	0.38
89	0.65	0.39	0.42	0.70	0.48	3.89	0.76	6.36	0.50
90	0.28	0.20	0.21	0.32	0.30	1.89	0.34	5.16	0.26
91	0.20	0.16	0.20	0.24	0.20	1.50	0.24	3.64	0.30
92	0.09	0.06	0.11	0.10	0.15	0.43	0.13	1.23	0.13
93	0.21	0.16	0.18	0.25	0.44	1.25	0.47	3.56	0.23
94	0.33	0.23	0.26	0.40	0.31	1.94	0.54	5.52	0.33
95	0.28	0.18	0.19	0.26	0.25	1.35	0.40	4.19	0.22
96	0.62	0.60	0.60	0.92	0.70	3.65	1.17	10.04	0.18
97	3.78	3.85	4.14	5.75	4.19	28.18	7.65	51.01	0.63
98	7.22	7.29	7.23	11.28	7.90	55.79	14.41	52.04	1.12
99	9.35	9.34	9.28	15.71	10.27	66.26	18.26	55.65	1.51
100	5.43	5.40	5.44	8.60	6.07	45.90	10.84	53.73	0.84

Table B.2.3: The time (in seconds) taken to analyse unreliability of the first phase in the mission alternative that is different from current mission, which in this case is the third phase, using the nine ordering schemes

Mission	scheme 1	scheme 2	scheme 3	scheme 4	scheme 5	scheme 6	scheme 7	scheme 8	BOI
1	0.02	0.03	0.02	0.11	0.03	0.14	0.09	0.42	0.03
2	1.59	1.62	0.55	0.88	0.41	1.53	2.35	3.25	1.57
3	41.84	58.89	23.35	10.56	12.47	46.68	68.10	20.57	4.03
4	26.39	38.94	8.84	10.13	5.45	17.54	112.37	11.35	67.37
5	11.69	15.39	4.27	6.36	2.46	8.17	41.54	8.18	42.94
6	0.29	0.30	0.18	0.30	0.19	0.50	0.54	2.80	0.10
7	0.82	1.23	1.06	1.40	0.62	1.37	1.44	12.25	1.16
8	17.05	14.70	8.83	9.78	3.96	4.67	12.15	21.07	11.72
9	17.96	29.72	14.50	17.22	7.43	24.59	31.54	23.73	3.41
10	23.94	25.84	14.34	16.99	7.23	16.43	27.88	22.64	11.89
11	17.55	18.04	11.07	10.59	5.08	15.74	19.35	22.56	10.04
12	1.99	2.72	1.58	2.70	0.98	2.09	6.65	13.27	0.53
13	0.04	0.06	0.03	0.07	0.04	0.23	0.11	0.66	0.01
14	5.77	26.24	8.33	3.28	4.01	4.78	7.74	9.39	34.05
15	28.19	423.27	82.53	18.57	24.06	26.08	431.43	21.51	137.17
16	0.67	1.42	0.70	0.67	0.58	1.09	0.92	6.21	0.43
17	0.70	0.61	0.42	0.78	0.82	1.83	1.32	5.71	0.50
18	10.85	7.90	6.51	7.39	7.12	8.84	7.37	15.52	3.62
19	16.76	16.49	10.61	10.24	10.80	18.32	17.86	19.84	0.41
20	15.86	12.44	8.20	7.84	5.99	11.55	13.58	17.44	3.36
21	2.10	1.39	1.02	2.49	2.18	3.01	5.75	9.97	0.15
22	0.16	0.16	0.14	0.51	0.19	0.52	0.73	2.70	0.02
23	0.79	1.07	1.13	2.29	1.03	1.41	1.97	3.84	0.28
24	2.00	2.60	2.28	3.65	2.00	2.83	3.70	5.51	0.21
25	4.50	4.86	4.41	6.81	3.39	4.89	6.73	6.53	0.95
26	1.28	1.67	1.79	2.78	1.42	2.01	2.65	4.45	0.53
27	0.31	0.28	0.25	0.65	0.27	0.57	0.91	3.58	0.10
28	0.09	0.23	0.21	0.15	0.11	0.43	0.23	2.12	0.06
29	19.06	309.62	199.90	60.03	8.33	14.64	199.98	66.98	173.84
30	0.06	0.05	0.05	0.12	0.05	0.22	0.19	0.51	0.02
31	1.15	3.71	1.38	2.20	0.64	2.13	2.17	2.95	0.74
32	3.15	5.38	3.07	4.34	1.58	5.85	6.06	4.15	3.15
33	2.42	7.39	2.12	2.76	0.81	2.97	8.18	3.69	1.82
34	1.84	5.88	1.79	2.66	0.71	3.71	6.58	3.43	1.65
35	0.56	0.75	0.45	0.91	0.37	1.14	0.93	2.58	0.93
36	0.03	0.05	0.05	0.11	0.05	0.18	0.16	0.21	0.01
37	1.31	1.06	0.95	1.66	0.95	1.99	1.62	3.66	0.23
38	5.17	2.76	2.44	4.77	2.66	3.98	3.66	6.62	2.09

39	10.61	5.79	4.93	10.09	9.18	8.39	7.23	9.84	2.53
40	1.85	1.30	1.22	2.32	1.11	1.92	1.95	4.28	0.88
41	0.58	0.43	0.47	0.83	0.48	0.85	1.18	2.94	0.60
42	0.04	0.04	0.04	0.10	0.05	0.18	0.13	0.46	0.02
43	0.79	0.98	0.84	1.76	0.64	1.45	1.43	3.85	0.38
44	2.38	2.74	2.46	4.32	2.32	3.95	3.65	7.03	1.19
45	5.17	5.37	4.55	7.91	4.13	6.66	6.40	10.53	1.31
46	1.16	1.52	1.47	2.37	1.18	2.20	2.16	4.71	0.71
47	0.46	0.31	0.32	0.77	0.40	0.76	0.90	3.06	0.37
48	0.11	0.12	0.16	0.38	0.11	0.28	0.19	0.80	0.12
49	5.62	4.50	3.43	6.63	3.09	4.29	4.26	13.52	2.44
50	0.03	0.05	0.04	0.11	0.03	0.19	0.14	0.50	0.09
51	0.70	3.85	4.08	2.84	0.82	1.64	4.69	2.90	4.60
52	2.40	13.73	15.44	6.29	1.93	5.17	15.28	5.74	7.82
53	5.25	32.64	38.21	11.93	3.99	10.48	34.09	9.14	9.71
54	2.02	6.18	6.16	3.69	1.18	2.20	6.64	3.80	5.87
55	0.43	0.74	0.84	0.92	0.30	1.02	0.68	2.41	1.94
56	0.05	0.05	0.04	0.10	0.04	0.19	0.16	0.39	0.04
57	0.76	0.97	1.08	2.79	0.42	1.33	1.63	2.18	0.89
58	1.63	3.49	2.65	4.24	1.92	2.85	4.19	2.97	1.68
59	2.51	6.80	5.68	6.31	1.63	4.37	7.72	3.56	1.84
60	0.87	2.01	1.88	3.71	0.86	2.11	2.68	2.73	1.30
61	0.40	0.27	0.28	0.79	0.51	0.66	0.84	1.43	0.30
62	0.04	0.06	0.07	0.16	0.08	0.16	0.17	0.89	0.04
63	1.53	0.79	0.82	2.54	0.65	1.07	1.37	3.93	1.03
64	3.46	2.51	2.12	5.32	1.92	2.89	3.29	5.45	2.60
65	6.11	4.55	3.65	9.29	4.21	8.86	5.91	7.51	3.54
66	2.22	1.34	1.17	3.24	1.05	1.50	2.02	4.14	2.39
67	0.56	0.32	0.36	0.91	0.36	0.74	0.91	2.89	0.66
68	0.02	0.03	0.03	0.10	0.05	0.16	0.12	0.58	0.01
69	0.64	0.39	0.38	0.88	0.32	0.85	0.87	2.93	0.11
70	1.43	0.83	0.86	1.50	0.63	1.53	1.55	3.52	0.10
71	2.21	1.57	1.57	2.44	0.90	2.49	2.54	3.90	0.03
72	0.98	0.47	0.55	1.05	0.33	1.10	1.07	3.25	0.04
73	0.21	0.15	0.16	0.41	0.16	0.50	0.70	2.39	0.03
74	0.03	0.03	0.03	0.11	0.03	0.15	0.13	0.40	0.02
75	0.71	1.74	1.59	1.09	0.39	1.12	2.22	1.93	1.01
76	2.34	4.75	4.36	2.31	0.90	2.62	5.51	3.05	4.24
77	4.29	9.12	8.38	3.98	1.70	4.22	10.19	4.33	1.02
78	1.09	2.72	2.51	1.47	0.43	1.67	3.36	2.13	0.45
79	0.41	0.41	0.45	0.48	0.25	0.58	0.71	1.74	0.68
80	0.05	0.05	0.05	0.06	0.06	0.23	0.08	0.76	0.03
81	0.68	0.32	0.33	0.51	0.72	1.14	0.69	1.50	0.09
82	3.00	0.85	0.89	1.12	1.29	2.10	1.10	2.68	0.20
83	8.20	1.72	1.72	1.92	2.65	3.94	1.97	4.22	0.14
84	0.82	0.43	0.43	0.56	1.03	1.17	0.83	1.49	0.08
85	0.49	0.20	0.21	0.37	0.39	0.75	0.39	1.30	0.12
86	0.14	0.10	0.12	0.13	0.18	0.56	0.09	2.05	0.20
87	2.21	1.43	1.46	2.19	1.97	4.23	2.12	4.60	0.42
88	5.36	3.76	3.57	4.41	4.02	6.85	3.52	7.33	0.71
89	10.71	6.46	6.65	8.21	7.18	13.93	7.02	9.73	0.75
90	3.51	2.07	2.06	2.40	2.57	6.40	1.01	5.95	0.43
91	1.07	0.68	0.78	1.44	1.19	2.27	0.37	3.97	0.42
92	0.09	0.06	0.11	0.10	0.15	0.43	0.13	1.23	0.13
93	1.66	1.12	1.09	0.92	2.52	2.49	2.13	4.00	0.20
94	8.40	5.48	5.66	2.85	4.50	5.09	5.95	8.20	0.48
95	2.68	1.81	1.63	1.10	2.41	3.44	1.10	5.46	0.19
96	0.62	0.60	0.60	0.92	0.70	3.65	1.17	10.04	0.18
97	7.08	7.24	7.62	8.86	7.16	31.27	11.15	54.38	0.92
98	19.18	19.52	19.43	22.55	18.79	67.22	26.50	63.57	1.70
99	32.86	33.43	33.23	39.99	31.52	87.21	42.36	78.73	2.14

100	12.38	12.23	12.32	15.08	12.26	52.56	17.67	60.11	1.19

Table B.2.4: The time for analysing unreliability of the first phase in the mission alternative that is different from current mission, which in this case is the penultimate phase, using the nine ordering schemes

B.3 Testing Results for Approximation Models

- When the time needed to approximate the lower, upper, or both bounds for the entire mission unreliability is longer than the exact mission unreliability analysis, the value is denoted as invalid and thus will show empty cells in Table B.3.1, Table B.3.2 and Table B.3.3.

- For all of these three tables, the bigger the truncation limit, the fewer invalid values there are.

Mission	*Analytic TL*	10^{-15}	10^{-13}	10^{-11}	10^{-09}	10^{-07}	10^{-05}	10^{-03}	10^{-01}
1								12.31%	10.77%
2									51.49%
3									53.99%
4	93.14%								48.23%
5									49.41%
6	94.88%								48.48%
7									47.71%
8	95.77%								43.49%
9					45.80%	49.39%	78.80%	94.74%	94.60%
10							47.94%	74.19%	74.25%
11						13.30%	61.84%	90.52%	90.82%
12							31.71%	81.14%	81.00%
13									28.44%
14									8.02%
15									26.51%
16							64.47%	64.91%	65.35%
17							51.07%	52.79%	52.58%
18							53.54%	53.02%	53.37%
19							49.14%	46.57%	50.57%
20							59.19%	58.99%	58.99%
21							53.99%	54.31%	54.31%
22								36.71%	89.45%
23									71.94%
24								62.09%	93.66%
25								17.14%	85.24%
26									49.18%
27									53.33%
28									
29									65.43%
30									29.41%
31									78.42%
32								11.78%	87.31%
33	70.00%	70.00%	66.67%	73.33%	73.33%	73.33%	73.33%	80.00%	80.00%
34									2.86%
35								76.74%	81.40%
36							13.55%	96.80%	99.50%
37								95.33%	99.90%

38								95.77%	99.90%
39								95.61%	99.89%
40								73.47%	95.50%
41						18.84%	46.89%	68.74%	71.94%
42					24.57%	87.06%	97.94%	99.60%	99.75%
43						85.61%	98.96%	99.84%	99.96%
44						84.35%	98.93%	99.84%	99.95%
45						84.44%	98.05%	99.72%	99.83%
46					5.42%	61.09%	85.35%	93.01%	94.55%
47									25.00%
48					54.83%	87.53%	95.85%	98.68%	99.49%
49					75.04%	97.69%	99.69%	99.96%	99.99%
50					43.44%	80.93%	92.37%	96.33%	97.95%
51					10.24%	32.20%	46.83%	59.51%	66.83%
52						52.40%	88.41%	95.87%	97.81%
53						2.29%	81.60%	94.42%	97.96%
54						30.91%	84.67%	94.77%	97.74%
55						36.31%	73.55%	82.47%	87.79%
56	12.96%	12.96%	14.81%	12.96%	11.11%	14.81%	25.93%	37.04%	48.15%
57							47.23%	79.16%	87.56%
58							48.91%	88.72%	95.95%
59							48.56%	94.80%	98.73%
60							50.46%	85.33%	92.35%
61							26.52%	52.27%	67.42%
62	10.71%				7.14%	17.86%	39.29%	42.86%	57.14%
63				54.64%	84.90%	96.20%	98.88%	99.58%	99.74%
64	52.94%			20.59%	41.18%	52.94%	61.76%	67.65%	73.53%
65				11.09%	66.95%	89.04%	95.93%	98.50%	99.44%
66					56.34%	90.31%	97.58%	99.40%	99.84%
67					57.52%	87.02%	95.60%	98.47%	99.47%
68				2.54%	65.03%	88.57%	95.94%	98.61%	99.50%
69	56.85%		12.78%	47.36%	71.77%	83.78%	92.65%	96.17%	97.93%
70	0.00%					6.67%	20.00%	26.67%	40.00%
71					48.72%	85.45%	95.21%	98.75%	99.52%
72					78.59%	97.23%	99.43%	99.87%	99.97%
73					88.62%	99.42%	99.93%	99.99%	100.00%
74					54.68%	88.40%	97.11%	99.40%	99.87%
75					42.50%	72.88%	89.50%	95.13%	96.00%
76	18.75%					18.75%	31.25%	43.75%	62.50%
77						60.30%	82.03%	92.55%	96.41%
78						55.88%	87.88%	97.51%	99.41%
79						64.71%	93.05%	99.09%	99.83%
80						64.88%	84.72%	94.78%	97.62%
81	3.83%				26.48%	63.41%	74.91%	84.32%	89.20%
82						15.61%	50.73%	76.59%	89.76%
83						77.68%	96.46%	99.48%	99.89%
84						4.00%	32.00%	44.00%	64.00%
85						45.33%	79.45%	94.37%	98.52%
86					8.92%	74.77%	95.17%	99.13%	99.84%
87					61.04%	94.31%	99.36%	99.93%	99.99%
88						54.26%	86.93%	96.75%	99.25%
89					10.74%	48.85%	74.68%	86.45%	94.37%
90	0.00%					4.35%		34.78%	60.87%
91					35.38%	77.40%	94.84%	98.79%	99.62%
92						64.33%	93.72%	98.85%	99.76%
93						41.41%	91.28%	98.73%	99.80%
94						76.44%	95.51%	99.20%	99.83%
95					21.95%	66.30%	87.27%	95.83%	98.02%
96							14.29%	40.48%	73.81%
97							50.29%	91.29%	98.56%
98							84.58%	98.98%	99.89%

99						35.27%	96.16%	99.82%	99.99%
100							61.76%	95.92%	99.44%
Average	42.49%	41.48%	31.42%	31.79%	45.44%	56.27%	71.51%	80.86%	80.64%

Table B.3.1: Percentage of time saved with different truncation limits for mission unreliability lower bound approximation using truncation method 3

Mission	*Analytic TL*	10^{-15}	10^{-13}	10^{-11}	10^{-09}	10^{-07}	10^{-05}	10^{-03}	10^{-01}
1								18.46%	16.92%
2									65.45%
3									70.59%
4	93.14%								62.17%
5									67.28%
6	94.88%								66.72%
7									67.16%
8	95.77%								67.07%
9					60.13%	62.47%	83.10%	96.46%	96.48%
10							27.63%	70.81%	70.94%
11						28.98%	60.64%	92.01%	92.01%
12						2.43%		81.29%	81.14%
13									42.20%
14									58.29%
15									44.58%
16							75.44%	75.88%	76.32%
17							74.46%	74.03%	74.46%
18							76.17%	76.51%	76.17%
19							71.71%	67.14%	69.71%
20							76.16%	75.96%	75.76%
21							71.88%	72.20%	72.84%
22								48.69%	92.07%
23									81.36%
24								71.57%	95.24%
25								37.62%	88.81%
26									63.93%
27									66.67%
28									2.33%
29									74.47%
30									38.24%
31									84.59%
32								39.09%	92.79%
33	70.00%	73.33%	70.00%	70.00%	73.33%	73.33%	73.33%	80.00%	80.00%
34								5.71%	5.71%
35								79.07%	83.72%
36							45.65%	99.61%	99.70%
37							43.24%	98.37%	99.98%
38							38.75%	98.90%	99.97%
39							16.32%	98.79%	99.96%
40								81.51%	97.11%
41						12.42%	46.89%	68.94%	71.94%
42					39.45%	89.91%	98.55%	99.69%	99.79%
43						90.33%	99.28%	99.91%	99.97%
44						90.46%	99.23%	99.91%	99.96%
45					11.53%	88.60%	98.71%	99.81%	99.86%
46					10.34%	61.99%	86.65%	93.91%	95.32%
47								0.00%	41.67%
48					58.97%	89.21%	96.58%	98.95%	99.54%
49					82.12%	98.60%	99.85%	99.98%	99.99%
50					46.70%	82.23%	93.35%	96.98%	98.14%
51					14.63%	33.66%	47.80%	60.49%	68.78%
52						57.97%	89.91%	96.39%	98.23%

53						18.20%	84.25%	95.93%	98.54%
54						39.63%	86.77%	95.67%	98.20%
55						38.97%	74.49%	83.41%	89.20%
56	14.81%	12.96%	14.81%	12.96%	12.96%	12.96%	25.93%	37.04%	50.00%
57							54.72%	82.31%	92.65%
58							63.95%	91.70%	98.78%
59							75.24%	96.49%	99.78%
60							62.32%	87.79%	96.49%
61							27.27%	53.03%	77.27%
62	17.86%				3.57%	17.86%	39.29%	42.86%	60.71%
63				55.75%	85.66%	96.53%	98.95%	99.64%	99.75%
64	55.88%			20.59%	41.18%	55.88%	61.76%	67.65%	73.53%
65				16.87%	70.27%	89.95%	96.64%	98.96%	99.56%
66					61.79%	91.56%	98.12%	99.64%	99.89%
67					61.80%	88.36%	96.43%	98.96%	99.64%
68				6.89%	67.06%	89.46%	96.64%	99.04%	99.62%
69	57.61%		14.23%	48.05%	72.69%	86.84%	93.11%	96.71%	98.24%
70	0.00%					0.00%	20.00%	26.67%	53.33%
71					58.83%	89.12%	96.50%	99.21%	99.66%
72					86.89%	98.16%	99.66%	99.94%	99.98%
73					94.48%	99.72%	99.96%	100.00%	100.00%
74					67.28%	92.28%	98.26%	99.70%	99.92%
75	0.50%				45.88%	77.25%	91.50%	95.75%	96.75%
76	18.75%					18.75%	31.25%	50.00%	68.75%
77					9.82%	62.05%	87.82%	94.65%	97.55%
78						69.61%	93.31%	98.85%	99.73%
79						77.87%	97.01%	99.67%	99.95%
80					11.74%	70.61%	90.96%	96.88%	98.60%
81					23.34%	63.76%	78.05%	86.06%	91.99%
82						16.59%	51.22%	77.07%	90.73%
83					3.85%	81.39%	97.32%	99.63%	99.92%
84						4.00%	32.00%	48.00%	68.00%
85					4.77%	50.97%	82.24%	95.77%	98.97%
86					29.25%	81.04%	96.65%	99.50%	99.92%
87					75.23%	96.75%	99.66%	99.97%	100.00%
88					9.38%	60.98%	89.95%	97.91%	99.58%
89					11.51%	49.10%	76.21%	87.72%	94.88%
90	4.35%					0.00%	21.74%	30.43%	73.91%
91					51.43%	85.56%	95.18%	99.20%	99.75%
92						76.66%	95.74%	99.59%	99.90%
93						60.20%	94.47%	99.66%	99.93%
94					35.13%	85.28%	96.95%	99.72%	99.93%
95					29.42%	65.75%	87.16%	96.27%	98.46%
96							16.67%	40.48%	76.19%
97							65.13%	94.50%	99.05%
98						20.44%	91.36%	99.58%	99.94%
99						70.04%	98.18%	99.95%	99.99%
100							77.49%	97.98%	99.70%
Average	43.63%	43.15%	33.02%	33.02%	43.50%	57.89%	74.04%	82.16%	84.67%

Table B.3.2: Percentage of time saved with different truncation limits for mission unreliability upper bound approximation using truncation method 3

Mission	*Analytic TL*	10^{-15}	10^{-13}	10^{-11}	10^{-09}	10^{-07}	10^{-05}	10^{-03}	10^{-01}
1									
2									16.93%
3									24.58%
4	86.28%								10.40%
5									16.69%
6	89.76%								15.20%
7									14.87%
8	91.55%								10.57%
9					5.93%	11.86%	61.90%	91.20%	91.08%
10								45.00%	45.19%
11							22.48%	82.53%	82.83%
12								62.43%	62.14%
13									
14									
15									
16							39.91%	40.79%	41.67%
17							25.54%	26.82%	27.04%
18							29.71%	29.53%	29.53%
19							20.86%	13.71%	20.29%
20							35.35%	34.95%	34.75%
21							25.88%	26.52%	27.16%
22									81.52%
23									53.31%
24								33.65%	88.90%
25									74.05%
26									13.11%
27									20.00%
28									
29									39.89%
30									
31									63.01%
32									80.09%
33	40.00%	43.33%	36.67%	43.33%	46.67%	46.67%	46.67%	60.00%	60.00%
34									
35								55.81%	65.12%
36								96.41%	99.20%
37								93.71%	99.88%
38								94.67%	99.86%
39								94.41%	99.85%
40								54.98%	92.60%
41								37.68%	43.89%
42						76.97%	96.49%	99.29%	99.54%
43						75.93%	98.24%	99.75%	99.93%
44						74.81%	98.16%	99.75%	99.91%
45						73.05%	96.76%	99.52%	99.69%
46						23.08%	72.00%	86.92%	89.86%
47									
48					13.79%	76.74%	92.43%	97.63%	99.03%
49					57.16%	96.28%	99.54%	99.94%	99.98%
50						63.16%	85.72%	93.30%	96.09%
51								20.00%	35.61%
52						10.37%	78.32%	92.26%	96.04%
53							65.85%	90.34%	96.50%
54							71.44%	90.44%	95.95%
55							48.04%	65.88%	77.00%
56									
57							1.95%	61.47%	80.21%
58							12.86%	80.42%	94.73%
59							23.80%	91.29%	98.51%
60							12.77%	73.12%	88.84%

61								5.30%	44.70%
62									17.86%
63				10.39%	70.56%	92.74%	97.83%	99.22%	99.49%
64	8.82%					8.82%	23.53%	35.29%	47.06%
65					37.23%	78.99%	92.57%	97.46%	99.00%
66					18.13%	81.87%	95.70%	99.05%	99.73%
67					19.32%	75.38%	92.03%	97.44%	99.12%
68					32.09%	78.02%	92.58%	97.65%	99.12%
69	14.46%				44.45%	70.62%	85.77%	92.88%	96.17%
70									
71					7.55%	74.57%	91.71%	97.96%	99.19%
72					65.48%	95.39%	99.09%	99.81%	99.95%
73					83.10%	99.14%	99.90%	99.98%	100.00%
74					21.96%	80.68%	95.37%	99.09%	99.79%
75						50.13%	81.00%	90.88%	92.75%
76									31.25%
77						22.35%	69.85%	87.20%	93.95%
78						25.49%	81.19%	96.36%	99.15%
79						42.58%	90.06%	98.77%	99.78%
80						35.49%	75.69%	91.66%	96.23%
81						27.18%	52.96%	70.38%	81.18%
82							1.95%	53.66%	80.49%
83						59.07%	93.77%	99.11%	99.81%
84									32.00%
85							61.69%	90.14%	97.49%
86						55.81%	91.83%	98.62%	99.76%
87					36.28%	91.06%	99.02%	99.90%	99.99%
88						15.23%	76.88%	94.65%	98.83%
89							50.90%	74.17%	89.26%
90									34.78%
91						62.96%	90.02%	98.00%	99.37%
92						40.99%	89.46%	98.44%	99.66%
93						1.61%	85.75%	98.39%	99.72%
94						61.72%	92.46%	98.93%	99.75%
95						32.05%	74.42%	92.10%	96.49%
96									50.00%
97							15.43%	85.79%	97.62%
98							75.93%	98.56%	99.83%
99						5.31%	94.34%	99.77%	99.98%
100							39.25%	93.91%	99.14%
Average	55.15%	43.33%	36.67%	26.86%	37.31%	53.93%	66.50%	80.02%	76.00%

Table B.3.3: Percentage of time saved with different truncation limits for both mission unreliability upper and lower bound approximations using truncation method 3

Mission	*Analytic TL*	10^{-15}	10^{-13}	10^{-11}	10^{-09}	10^{-07}	10^{-05}	10^{-03}	10^{-01}
1	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.96%	99.96%
2	100.00%		100.00%	100.00%	99.95%	99.50%	98.67%	97.68%	92.94%
3	100.00%		100.00%	100.00%	99.96%	99.64%	98.97%	98.33%	95.09%
4		100.00%	100.00%	100.00%	99.95%	99.41%	98.70%	97.33%	92.57%
5				99.99%	99.94%	99.35%	98.43%	97.06%	90.41%
6			100.00%	99.99%	99.95%	99.72%	99.13%	97.05%	88.16%
7					99.98%	99.82%	98.78%	98.02%	91.53%
8					99.94%	99.69%	98.92%	97.34%	90.46%
9		100.00%	100.00%	100.00%	99.99%	99.99%	99.96%	99.93%	99.93%
10	100.00%	100.00%	100.00%	100.00%	99.99%	99.99%	89.79%	89.77%	89.77%
11		100.00%	99.99%	99.99%	99.95%	99.94%	73.54%	73.43%	73.43%
12	100.00%	100.00%	100.00%	100.00%	99.99%	99.99%	81.17%	81.16%	81.16%
13	100.00%	100.00%	100.00%	100.00%	100.00%	99.98%	99.24%	97.12%	85.30%
14	100.00%	100.00%	100.00%	100.00%	99.99%	99.91%	99.12%	93.85%	77.92%

15	100.00%	100.00%	100.00%	100.00%	100.00%	99.96%	99.42%	98.21%	90.79%
16	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
17		100.00%	100.00%	100.00%	100.00%	100.00%	97.04%	97.04%	97.04%
18		100.00%	100.00%	100.00%	100.00%	100.00%	96.68%	96.68%	96.68%
19		100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
20		100.00%	100.00%	100.00%	100.00%	100.00%	96.48%	96.48%	96.48%
21	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	96.71%	96.71%	96.71%
22	99.80%	99.89%	99.30%	90.75%	85.72%	79.93%	70.03%	50.34%	6.36%
23	100.00%	99.96%	99.91%	99.41%	96.94%	96.10%	94.09%	65.74%	21.52%
24	99.99%	99.98%	99.58%	99.37%	98.31%	85.87%	83.49%	50.94%	7.89%
25	100.00%	100.00%	100.00%	100.00%	99.97%	98.40%	80.68%	73.45%	33.48%
26	100.00%	100.00%	100.00%	100.00%	99.95%	91.35%	88.29%	58.37%	14.62%
27	100.00%	100.00%	100.00%	100.00%	100.00%	99.94%	98.06%	83.44%	28.59%
28	100.00%	100.00%	100.00%	100.00%	99.95%	98.38%	88.81%	77.47%	20.50%
29	100.00%	99.98%	98.97%	98.67%	97.59%	93.78%	75.80%	47.55%	7.98%
30	100.00%	100.00%	100.00%	99.99%	99.87%	98.24%	81.88%	68.18%	60.78%
31	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.97%	99.23%	89.76%
32	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.90%	98.34%	87.87%
33	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
34	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.98%	99.98%
35	100.00%	100.00%	100.00%	100.00%	100.00%	99.98%	85.87%	88.22%	70.36%
36							94.10%	69.55%	3.86%
37								67.31%	6.11%
38								77.99%	7.99%
39								56.66%	0.55%
40	100.00%	100.00%	100.00%	100.00%	100.00%	99.69%	91.25%	68.45%	28.22%
41		100.00%	100.00%	100.00%	100.00%	99.79%	84.71%	69.02%	66.56%
42			100.00%	100.00%	99.91%	97.32%	86.71%	40.80%	1.37%
43						99.78%	96.03%	79.79%	7.94%
44						99.87%	97.42%	88.36%	0.32%
45					99.98%	99.64%	93.14%	87.97%	4.37%
46		100.00%	100.00%	100.00%	99.96%	98.50%	55.07%	65.16%	1.26%
47	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.95%	99.15%	15.37%
48	100.00%	100.00%	100.00%	100.00%	99.99%	99.94%	98.84%	18.44%	0.29%
49					99.97%	99.54%	96.46%	74.26%	0.38%
50	100.00%	100.00%	100.00%	100.00%	100.00%	99.94%	99.06%	97.30%	12.97%
51	100.00%	100.00%	100.00%	100.00%	100.00%	99.99%	99.68%	98.51%	7.64%
52	100.00%	100.00%	100.00%	100.00%	100.00%	99.88%	96.95%	62.55%	8.78%
53	100.00%	100.00%	100.00%	100.00%	100.00%	99.91%	98.15%	80.46%	20.43%
54	100.00%	100.00%	100.00%	100.00%	100.00%	99.95%	98.84%	79.33%	1.70%
55	100.00%	100.00%	100.00%	100.00%	100.00%	99.98%	99.41%	83.25%	14.02%
56	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.99%	99.77%	0.02%
57	100.00%	100.00%	100.00%	100.00%	99.99%	99.56%	93.60%	69.63%	0.06%
58	100.00%	100.00%	100.00%	100.00%	100.00%	99.91%	96.95%	75.45%	0.07%
59	100.00%	100.00%	100.00%	100.00%	100.00%	99.90%	96.99%	88.64%	4.44%
60	100.00%	100.00%	100.00%	100.00%	100.00%	99.95%	99.07%	86.87%	2.94%
61	100.00%	100.00%	100.00%	100.00%	100.00%	99.94%	96.40%	90.92%	57.28%
62	100.00%	100.00%	100.00%	100.00%	100.00%	99.99%	99.98%	99.28%	0.01%
63	100.00%	100.00%	100.00%	100.00%	99.98%	99.53%	67.88%	23.50%	3.15%
64	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.94%	99.92%	0.00%
65	100.00%	100.00%	100.00%	100.00%	99.98%	99.83%	91.53%	78.19%	2.98%
66	100.00%	100.00%	100.00%	100.00%	99.99%	99.87%	98.21%	89.80%	0.21%
67	100.00%	100.00%	100.00%	100.00%	100.00%	99.96%	99.19%	95.31%	0.43%
68	100.00%	100.00%	100.00%	100.00%	100.00%	99.97%	98.97%	94.46%	0.09%
69	100.00%	100.00%	100.00%	100.00%	100.00%	99.97%	99.10%	97.42%	0.13%
70	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.98%	95.53%	2.35%
71	100.00%	100.00%	100.00%	100.00%	99.98%	99.73%	92.67%	35.12%	10.15%
72				100.00%	99.99%	99.92%	98.51%	79.13%	0.65%
73					99.77%	83.38%	13.72%	3.85%	0.36%
74	100.00%	100.00%	100.00%	100.00%	99.99%	99.70%	92.28%	32.38%	0.56%
75	100.00%	100.00%	100.00%	100.00%	99.98%	99.74%	93.54%	30.41%	0.00%

76	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.96%	98.39%	0.01%
77	100.00%	100.00%	100.00%	100.00%	99.99%	99.82%	95.95%	42.25%	0.01%
78	100.00%	100.00%	100.00%	100.00%	99.98%	99.78%	97.48%	69.18%	0.91%
79	100.00%			100.00%	99.96%	99.64%	96.18%	73.13%	0.10%
80	100.00%	100.00%	100.00%	100.00%	100.00%	99.90%	98.53%	83.79%	0.19%
81	99.99%	100.00%	100.00%	100.00%	99.97%	99.52%	93.83%	37.74%	1.33%
82	100.00%	100.00%	100.00%	100.00%	100.00%	99.93%	97.80%	80.45%	21.99%
83		100.00%	100.00%	100.00%	99.95%	98.45%	87.75%	70.21%	92.00%
84	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.99%	99.99%	0.12%
85	100.00%	100.00%	100.00%	100.00%	99.98%	99.73%	98.44%	94.35%	2.48%
86	100.00%	100.00%	100.00%	100.00%	99.99%	99.78%	99.56%	98.61%	0.74%
87					100.00%	99.89%	99.73%	99.09%	66.39%
88	100.00%	100.00%	100.00%	100.00%	100.00%	99.96%	99.88%	99.43%	88.07%
89	100.00%	100.00%	100.00%	100.00%	100.00%	99.97%	99.75%	97.41%	3.09%
90	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	99.97%	99.85%	27.50%
91	100.00%	100.00%	100.00%	99.99%	99.43%	93.18%	75.83%	34.93%	0.07%
92	100.00%	100.00%	100.00%	100.00%	99.89%	98.75%	94.37%	85.07%	0.15%
93	100.00%		100.00%	100.00%	99.99%	99.66%	96.74%	88.47%	0.14%
94	100.00%	100.00%	100.00%	100.00%	99.93%	99.04%	95.64%	88.80%	1.40%
95	100.00%	100.00%	100.00%	100.00%	99.99%	99.71%	90.38%	79.00%	0.11%
96	99.99%	100.00%	100.00%	100.00%	100.00%	99.99%	99.94%	95.94%	0.05%
97	100.00%	100.00%	100.00%	100.00%	99.78%	97.30%	81.88%	30.79%	0.52%
98					99.97%	99.30%	96.13%	89.54%	0.22%
99						98.33%	89.62%	85.90%	0.16%
100	100.00%	100.00%	100.00%	100.00%	99.99%	99.58%	97.55%	92.48%	0.19%
Average	100.00%	100.00%	99.98%	99.88%	99.77%	99.02%	94.02%	80.91%	30.18%

Table B.3.4: Accuracy of lower bound approximated using truncation method 3 compared with exact mission unreliability

Mission	*Analytic TL*	10^{-15}	10^{-13}	10^{-11}	10^{-09}	10^{-07}	10^{-05}	10^{-03}	10^{-01}
1	100.00%	100.00%	100.00%	100.00%	100.00%	98.41%	96.80%	74.40%	74.40%
2	99.99%	100.00%	100.00%	99.99%	99.82%	97.84%	95.07%	90.75%	76.29%
3	99.99%	100.00%	100.00%	99.98%	99.78%	97.27%	94.02%	88.28%	72.84%
4	47.01%	100.00%	100.00%	99.98%	99.83%	97.57%	95.07%	89.68%	76.95%
5	99.99%			99.98%	99.79%	97.77%	95.09%	91.00%	78.38%
6	50.90%		100.00%	99.95%	99.81%	99.08%	96.35%	90.84%	74.30%
7	100.00%			99.99%	99.93%	99.15%	95.17%	91.60%	75.96%
8	54.80%			99.94%	99.76%	98.73%	95.63%	89.26%	74.67%
9	99.51%	96.64%	96.46%	96.46%	75.12%	75.11%	59.50%	42.29%	42.29%
10	98.37%	99.16%	98.37%	98.37%	83.82%	83.81%	57.61%	47.17%	47.17%
11	100.00%	99.79%	93.59%	92.54%	64.32%	61.45%	47.96%	26.37%	26.37%
12	100.00%	100.00%	100.00%	100.00%	99.73%	96.73%	95.83%	66.40%	66.40%
13	100.00%	100.00%	100.00%	100.00%	99.99%	99.73%	95.52%	91.13%	80.84%
14	100.00%	100.00%	100.00%	100.00%	99.95%	99.61%	96.84%	93.52%	81.46%
15	100.00%	100.00%	100.00%	100.00%	99.88%	99.51%	93.12%	88.85%	70.36%
16	100.00%	99.51%	99.51%	98.49%	98.49%	98.49%	90.81%	90.81%	90.81%
17		99.73%	99.73%	99.16%	99.16%	99.16%	91.22%	91.22%	91.22%
18		99.76%	99.76%	99.27%	99.27%	99.27%	91.00%	91.00%	91.00%
19	100.00%	99.82%	99.82%	99.31%	99.31%	99.31%	91.04%	91.04%	91.04%
20		99.84%	99.84%	99.38%	99.38%	99.38%	90.88%	90.88%	90.88%
21	100.00%	99.44%	99.44%	98.63%	98.63%	98.63%	90.73%	90.73%	90.73%
22	98.90%	99.47%	97.40%	76.61%	61.06%	53.59%	41.59%	21.89%	14.74%
23	100.00%	99.63%	99.07%	97.48%	87.18%	85.23%	65.77%	32.76%	14.35%
24	99.61%	99.59%	97.63%	94.89%	91.49%	86.21%	61.46%	30.81%	24.00%
25	100.00%	100.00%	100.00%	100.00%	99.88%	97.82%	73.23%	59.90%	37.84%
26	100.00%	100.00%	100.00%	100.00%	99.99%	99.06%	79.22%	48.33%	31.88%
27	100.00%	100.00%	100.00%	100.00%	99.99%	99.65%	93.76%	67.83%	30.51%
28	100.00%	100.00%	100.00%	100.00%	99.99%	99.74%	83.04%	50.59%	32.16%
29	99.99%	99.95%	98.13%	96.27%	93.85%	88.20%	60.95%	31.20%	25.16%

30	100.00%	100.00%	100.00%	99.98%	99.63%	96.82%	78.24%	58.24%	37.64%
31	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%			
32	100.00%	100.00%	100.00%	100.00%	100.00%				
33	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	98.77%	95.44%	95.44%
34	100.00%	100.00%	100.00%	100.00%	99.22%	99.22%	97.36%	75.30%	75.30%
35	99.96%	100.00%	100.00%	100.00%	100.00%	99.51%	50.29%	5.54%	5.54%
36							73.28%	19.50%	19.50%
37								68.86%	48.81%
38								68.52%	54.14%
39								52.00%	40.62%
40	99.98%	100.00%	100.00%	100.00%	99.99%	97.97%	62.10%	22.99%	6.72%
41		100.00%	100.00%	100.00%	99.81%	91.74%	9.68%	0.37%	0.27%
42		100.00%	99.97%	99.35%	80.83%	37.19%	4.88%	0.22%	0.04%
43					98.60%	88.79%	37.55%	3.68%	0.50%
44					99.11%	92.67%	48.31%	6.12%	0.78%
45				99.93%	97.91%	84.91%	21.33%	2.25%	0.44%
46		100.00%	99.97%	99.14%	79.09%	38.02%	4.36%	0.30%	0.03%
47	99.83%	100.00%	100.00%	100.00%	100.00%	99.82%	83.26%	3.42%	0.94%
48	99.84%	100.00%	100.00%	99.99%	99.54%	82.36%	10.65%	0.78%	0.08%
49					99.21%	80.09%	12.48%	1.56%	0.38%
50	99.51%	100.00%	100.00%	99.99%	99.81%	94.62%	32.20%	1.89%	0.22%
51	99.96%	100.00%	100.00%	100.00%	99.97%	97.70%	59.23%	3.24%	0.12%
52	100.00%	100.00%	100.00%	99.99%	99.76%	92.27%	20.34%	1.42%	0.30%
53	100.00%	100.00%	100.00%	100.00%	99.91%	96.20%	45.21%	4.98%	1.01%
54	100.00%	100.00%	100.00%	100.00%	99.92%	97.38%	46.99%	4.87%	1.01%
55	100.00%	100.00%	100.00%	100.00%	99.99%	99.01%	72.95%	8.59%	0.38%
56	99.99%	100.00%	100.00%	100.00%	100.00%	99.99%	96.52%	22.65%	0.02%
57	99.85%	100.00%	100.00%	100.00%	99.81%	80.25%	16.73%	0.99%	0.08%
58	99.99%	100.00%	100.00%	100.00%	99.97%	96.89%	49.32%	4.10%	0.75%
59	100.00%	100.00%	100.00%	100.00%	99.98%	96.76%	54.86%	9.12%	2.11%
60	99.98%	100.00%	100.00%	100.00%	99.98%	97.27%	59.67%	4.36%	0.75%
61	99.85%	100.00%	100.00%	100.00%	99.99%	96.32%	29.32%	1.08%	0.06%
62	99.97%	100.00%	100.00%	100.00%	100.00%	99.90%	87.52%	9.57%	0.26%
63	99.97%	100.00%	100.00%	99.99%	99.58%	80.43%	5.52%	0.35%	0.07%
64	99.55%	100.00%	100.00%	100.00%	99.99%	99.43%	88.00%	20.49%	0.44%
65	99.91%	100.00%	100.00%	99.99%	99.31%	89.02%	17.33%	1.90%	0.19%
66	99.99%	100.00%	100.00%	99.99%	99.80%	94.42%	42.25%	5.67%	0.79%
67	99.99%	100.00%	100.00%	100.00%	99.96%	98.79%	69.74%	14.15%	2.23%
68	99.99%	100.00%	100.00%	100.00%	99.94%	98.47%	63.82%	11.79%	1.62%
69	99.84%	100.00%	100.00%	100.00%	99.68%	92.63%	30.48%	2.81%	0.13%
70	99.99%	100.00%	100.00%	100.00%	100.00%	99.99%	98.36%	69.31%	4.80%
71	99.99%	100.00%	100.00%	99.98%	99.61%	88.52%	33.55%	3.07%	0.43%
72			100.00%	99.99%	99.82%	95.27%	55.71%	7.96%	1.66%
73					98.82%	53.85%	11.34%	4.10%	3.95%
74	99.99%	100.00%	100.00%	99.98%	99.62%	88.11%	32.41%	3.51%	0.44%
75	99.99%	100.00%	100.00%	100.00%	99.64%	89.00%	38.60%	3.59%	0.39%
76	99.71%	100.00%	100.00%	100.00%	100.00%	99.84%	95.28%	44.18%	1.41%
77	99.84%	100.00%	100.00%	99.97%	99.22%	75.49%	17.60%	1.65%	0.14%
78	99.95%	100.00%	100.00%	99.98%	99.40%	85.84%	30.58%	3.06%	0.40%
79	99.98%	100.00%	100.00%	99.99%	99.29%	91.96%	43.90%	8.30%	1.35%
80	99.95%	100.00%	100.00%	99.99%	99.77%	90.54%	40.59%	5.32%	0.46%
81	99.28%	100.00%	100.00%	99.88%	97.01%	45.86%	6.68%	0.58%	0.03%
82	99.99%	100.00%	100.00%	100.00%	99.84%	94.64%	26.75%	2.68%	0.15%
83	100.00%	100.00%	100.00%	99.97%	98.36%	54.01%	5.75%	0.73%	0.17%
84	99.99%	100.00%	100.00%	100.00%	100.00%	99.85%	82.94%	6.17%	0.21%
85	99.99%	100.00%	100.00%	99.98%	99.10%	73.14%	4.21%	0.40%	0.02%
86	100.00%	100.00%	100.00%	99.99%	99.76%	86.21%	38.09%	5.31%	0.48%
87					99.81%	93.40%	55.99%	11.55%	1.66%
88	100.00%	100.00%	100.00%	100.00%	99.93%	95.49%	49.53%	8.46%	0.60%
89	99.99%	100.00%	100.00%	100.00%	99.96%	97.89%	42.09%	2.40%	0.13%
90	99.82%	100.00%	100.00%	100.00%	99.98%	99.54%	89.95%	25.43%	0.41%

91	99.97%	100.00%	100.00%	99.68%	88.58%	21.25%	3.34%	0.30%	0.05%
92	99.99%	100.00%	100.00%	99.96%	98.09%	71.67%	18.50%	1.34%	0.28%
93	100.00%	100.00%	100.00%	99.99%	99.83%	93.43%	39.42%	2.90%	1.05%
94	100.00%	100.00%	100.00%	99.98%	98.95%	76.75%	23.98%	1.23%	0.33%
95	99.87%	100.00%	100.00%	99.69%	98.94%	65.56%	4.42%	0.15%	0.01%
96	99.37%	100.00%	100.00%	100.00%	99.94%	98.85%	71.58%	3.09%	0.09%
97	99.77%	100.00%	100.00%	99.99%	97.32%	58.02%	2.92%	0.26%	0.08%
98	99.97%	100.00%	100.00%	99.99%	99.44%	79.39%	10.02%	0.72%	0.30%
99						74.06%	9.11%	2.07%	1.10%
100	99.98%	100.00%	100.00%	100.00%	99.65%	79.57%	11.07%	1.25%	0.36%
Average	98.32%	99.92%	99.78%	99.45%	97.61%	88.30%	51.66%	26.94%	19.92%

Table B.3.5: Accuracy of upper bound approximated using truncation method 3 compared with exact mission unreliability

TL	time P1	time P2	time P3	time P4	time P5	time P6	time P7	Total Time	M Unrelib	M BDD Size
1.00E-15	0.030	0.100	0.510	0.790	0.290	1.360	2.840	5.920	0.312905	8711
3.16E-15	0.030	0.090	0.450	0.740	0.310	1.412	2.653	5.685	0.312905	8711
1.00E-14	0.020	0.090	0.470	0.750	0.290	1.370	2.690	5.680	0.312905	8711
3.16E-14	0.030	0.090	0.460	0.740	0.290	1.340	2.734	5.684	0.312905	8711
1.00E-13	0.020	0.100	0.470	0.770	0.280	1.400	2.700	5.740	0.312905	8711
3.16E-13	0.030	0.050	0.350	0.550	0.260	1.260	2.392	4.892	0.312905	8185
1.00E-12	0.020	0.050	0.360	0.610	0.270	1.220	1.810	4.340	0.312905	8069
3.16E-12	0.020	0.050	0.220	0.400	0.250	1.070	1.430	3.440	0.312905	7439
1.00E-11	0.020	0.050	0.200	0.330	0.262	0.680	0.700	2.242	0.312905	6660
3.16E-11	0.030	0.050	0.180	0.240	0.200	0.540	0.560	1.800	0.312905	6167
1.00E-10	0.030	0.040	0.190	0.240	0.120	0.340	0.370	1.330	0.312905	5432
3.16E-10	0.020	0.050	0.190	0.230	0.130	0.340	0.360	1.320	0.312905	5432
1.00E-09	0.020	0.050	0.190	0.230	0.130	0.340	0.360	1.320	0.312905	5432
3.16E-09	0.020	0.050	0.190	0.240	0.120	0.340	0.360	1.320	0.312905	5432
1.00E-08	0.020	0.050	0.190	0.240	0.120	0.340	0.360	1.320	0.312905	5432
3.16E-08	0.030	0.040	0.190	0.240	0.120	0.340	0.370	1.330	0.312905	5432
1.00E-07	0.020	0.050	0.190	0.240	0.120	0.350	0.360	1.330	0.312905	5432
3.16E-07	0.030	0.050	0.190	0.230	0.120	0.350	0.360	1.330	0.312905	5440
1.00E-06	0.030	0.010	0.123	0.130	0.121	0.300	0.220	0.934	0.312903	5102
3.16E-06	0.020	0.020	0.110	0.130	0.120	0.240	0.230	0.870	0.312903	5009
1.00E-05	0.020	0.020	0.040	0.050	0.030	0.060	0.040	0.260	0.312879	4594
3.16E-05	0.030	0.010	0.050	0.040	0.030	0.060	0.050	0.270	0.312879	4594
1.00E-04	0.030	0.020	0.040	0.040	0.030	0.060	0.050	0.270	0.312879	4594
0.000316	0.030	0.010	0.040	0.050	0.030	0.060	0.040	0.260	0.312879	4594
1.00E-03	0.030	0.020	0.040	0.041	0.030	0.060	0.050	0.271	0.312879	4594
0.003162	0.030	0.010	0.050	0.040	0.030	0.060	0.050	0.270	0.312879	4594
1.00E-02	0.020	0.020	0.040	0.040	0.030	0.060	0.050	0.260	0.312879	4594
0.031623	0.030	0.010	0.040	0.050	0.030	0.060	0.040	0.260	0.312879	4594
1.00E-01	0.030	0.010	0.040	0.050	0.020	0.060	0.050	0.260	0.312879	4594
0.316228	0.020	0.020	0.040	0.040	0.030	0.060	0.050	0.260	0.312879	4594

Table B.3.6: Analysis data (time needed to approximate unreliability of mission phases, time needed to approximate the entire mission unreliability, the approximated lower bound probability for mission unreliability, the size of the truncated BDD representing the mission failure) for the example mission with varying truncation limit

	Mission Unreliability				Analysis time(s)				BDD sizes		
Mission	M1	M2	M3	EXACT	M1	M2	M3	EXACT	M2	M3	EXACT
1	1.000	0.001	0.005	0.001	0.059	0.122	0.549	2.019	2953	3478	5874
2	0.999	0.006	0.005	0.005	0.013	0.044	1.127	9.169	2260	3630	9912
3	1.000	0.008	0.009	0.007	0.012	0.042	1.382	4.095	1910	3796	7251
4	1.000	0.003	0.028	0.003	0.026	0.088	8.178	91.656	2658	9123	36085
5	0.999	0.001	0.016	0.001	0.013	0.036	0.214	2.474	1962	2235	6108

Table B.3.7: Analysis data for example missions using different upper bound approximation methods

	Analysis Time			Unreliability		
Mission	Method 2	Exact	Time Reduction	Method 2	Exact	Accuracy
1	0.05	0.08	41.03%	0.0505	0.0489	96.75%
2	0.36	0.44	17.70%	0.4091	0.1349	32.97%
3	0.33	0.50	34.27%	0.4083	0.1336	32.72%
4	0.34	0.45	24.28%	0.4092	0.1349	32.97%
5	0.50	0.66	23.66%	0.5670	0.1593	28.09%
6	0.53	0.62	14.90%	0.5911	0.1828	30.93%
7	0.86	0.87	1.83%	0.7505	0.1834	24.44%
8	0.84	0.91	6.85%	0.7973	0.2059	25.83%
9	0.62	10.75	94.19%	0.0226	0.0112	49.35%
10	0.86	1.61	46.51%	0.0382	0.0129	33.74%
11	0.70	5.43	87.07%	0.0186	0.0096	51.56%
12	0.25	0.84	70.43%	1.0000	0.6352	63.52%
13	0.11	0.13	10.94%	0.4413	0.3197	72.45%
14	0.23	0.20	-15.27%	0.5998	0.4592	76.56%
15	0.08	0.08	0.00%	0.3398	0.2256	66.38%
16	0.11	0.11	0.91%	0.4087	0.2898	70.90%
17	0.28	0.23	-20.09%	0.9794	0.4588	46.85%
18	0.39	0.28	-38.79%	1.0000	0.5035	50.35%
19	0.23	0.19	-24.47%	0.5456	0.4106	75.24%
20	0.30	0.23	-26.92%	0.5854	0.4435	75.75%
21	0.20	0.17	-18.71%	0.5054	0.3759	74.37%
22	0.12	1.16	89.33%	0.0003	0.0003	87.90%
23	0.13	0.50	73.69%	0.0004	0.0003	62.79%
24	0.12	1.98	93.83%	0.0005	0.0004	83.77%
25	0.05	0.43	87.59%	0.0001	0.0000	91.87%
26	0.05	0.12	62.81%	0.0000	0.0000	75.04%
27	0.09	0.20	54.90%	0.0006	0.0003	46.35%
28	0.04	0.04	-2.33%	0.0001	0.0000	55.88%
29	0.12	0.36	66.21%	0.0003	0.0003	83.16%
30	0.04	0.07	38.24%	0.0004	0.0003	81.94%
31	0.11	0.34	67.93%	0.7456	0.6771	90.82%
32	0.28	0.97	71.04%	1.0000	0.7532	75.32%
33	0.00	0.00	0.00%	0.1365	0.1353	99.11%
34	0.05	0.03	-43.75%	0.4186	0.2531	60.47%
35	0.00	0.00		0.0006	0.0006	99.25%
36	0.02	0.86	98.14%	0.0026	0.0023	87.35%
37	0.05	10.28	99.54%	0.0046	0.0040	87.85%
38	0.03	7.80	99.60%	0.0065	0.0050	76.04%
39	0.03	5.49	99.44%	0.0039	0.0035	90.49%
40	0.02	0.14	88.57%	0.0005	0.0005	99.03%
41	0.19	0.47	60.04%	0.0001	0.0001	93.23%
42	0.31	103.58	99.70%	0.0010	0.0008	82.63%
43	0.42	862.47	99.95%	0.0055	0.0043	77.66%
44	0.39	857.00	99.95%	0.0079	0.0054	68.05%
45	0.34	147.51	99.77%	0.0041	0.0032	77.97%
46	0.17	3.07	94.55%	0.0007	0.0006	92.14%
47	0.01	0.01	41.67%	0.0012	0.0011	98.09%
48	0.06	9.18	99.36%	0.0049	0.0042	86.98%

49	0.13	1018.38	99.99%	0.0082	0.0069	83.83%
50	0.05	2.18	97.71%	0.0009	0.0009	97.57%
51	0.11	0.23	53.91%	0.0001	0.0001	97.90%
52	0.15	6.90	97.90%	0.0008	0.0007	84.68%
53	0.19	10.69	98.24%	0.0037	0.0032	85.89%
54	0.15	6.78	97.74%	0.0034	0.0030	87.77%
55	0.07	0.68	89.75%	0.0003	0.0003	95.19%
56	0.03	0.04	31.71%	0.0001	0.0001	98.23%
57	0.07	0.66	89.53%	0.0007	0.0006	87.72%
58	0.10	4.43	97.65%	0.0033	0.0028	84.83%
59	0.18	38.34	99.53%	0.0098	0.0070	71.27%
60	0.08	1.44	94.52%	0.0023	0.0019	83.82%
61	0.04	0.13	73.48%	0.0003	0.0003	96.05%
62	0.01	0.03	58.62%	0.0001	0.0001	95.05%
63	0.35	113.79	99.69%	0.0009	0.0008	82.81%
64	0.01	0.03	75.76%	0.0001	0.0001	97.97%
65	0.04	7.04	99.43%	0.0007	0.0006	87.13%
66	0.07	44.99	99.85%	0.0028	0.0025	91.21%
67	0.05	10.41	99.55%	0.0061	0.0041	67.56%
68	0.04	9.48	99.54%	0.0031	0.0028	93.24%
69	0.03	1.37	98.03%	0.0001	0.0001	98.99%
70	0.01	0.02	62.50%	0.0029	0.0028	96.69%
71	0.04	8.57	99.50%	0.0037	0.0033	88.90%
72	0.09	273.97	99.97%	0.0316	0.0274	86.67%
73	0.16	7712.40	100.00%	0.0852	0.0575	67.53%
74	0.06	35.73	99.85%	0.0184	0.0176	95.92%
75	0.03	0.77	95.99%	0.0026	0.0025	97.14%
76	0.01	0.02	66.67%	0.0003	0.0003	99.65%
77	0.05	1.41	96.61%	0.0013	0.0011	87.12%
78	0.09	16.12	99.47%	0.0027	0.0023	86.30%
79	0.15	124.99	99.88%	0.0102	0.0074	72.22%
80	0.05	2.20	97.82%	0.0027	0.0024	89.02%
81	0.05	0.29	82.70%	0.0002	0.0002	97.62%
82	0.02	0.28	93.14%	0.0001	0.0001	97.93%
83	0.09	69.48	99.87%	0.0039	0.0024	60.67%
84	0.01	0.02	70.83%	0.0004	0.0004	96.04%
85	0.04	2.81	98.44%	0.0032	0.0027	84.60%
86	0.07	50.37	99.86%	0.0052	0.0046	88.66%
87	0.11	2045.74	99.99%	0.0288	0.0203	70.71%
88	0.05	6.05	99.16%	0.0057	0.0051	89.75%
89	0.02	0.39	93.86%	0.0003	0.0003	97.11%
90	0.01	0.02	75.00%	0.0002	0.0002	97.95%
91	0.03	7.28	99.57%	0.0009	0.0007	83.88%
92	0.07	18.47	99.64%	0.0053	0.0049	92.03%
93	0.09	37.66	99.77%	0.0249	0.0183	73.41%
94	0.04	24.55	99.82%	0.0062	0.0059	94.50%
95	0.02	0.90	98.12%	0.0002	0.0002	98.45%
96	0.01	0.04	80.49%	0.0001	0.0001	99.70%
97	0.04	4.05	98.96%	0.0003	0.0002	93.43%
98	0.08	91.40	99.92%	0.0035	0.0033	94.57%
99	0.13	1182.16	99.99%	0.0153	0.0117	76.57%
100	0.05	14.40	99.68%	0.0042	0.0040	96.13%
Average			72.29%			80.68%

Table B.3.8: Comparison of approximation method 2 to exact analysis

References

[1] J. D. Andrews. Component contributions to the failure of systems undergoing phased missions. In *17th Advances in Risk and Reliability Technology Symposium*, pages 155–168, 2007.

1 citation(s) on 1 page(s): 69.

[2] J. D. Andrews and T. R. Moss. *Reliability and risk assessment*. Wiley-Blackwell, 2nd edition, 2002.

7 citation(s) on 6 page(s): 16, 22, 25, 26, 38, and 74.

[3] J. D. Andrews, J. Poole, and W. H. Chen. Fast mission reliability prediction for Unmanned Aerial Vehicles. *Reliability Engineering and System Safety*, 120:3–9, 2013.

12 citation(s) on 6 page(s): 39, 51, 56, 58, 65, and 75.

[4] J. D. Andrews, D. R. Prescott, and R. Remenyte-Prescott. A systems reliability approach to decision making in autonomous multi-platform systems operating a phased mission. In *Annual Reliability and Maintainability Symposium*, pages 8–14. IEEE, 2008.

4 citation(s) on 4 page(s): 64, 65, 66, and 69.

[5] J. D. Andrews, R. Remenyte-Prescott, and C. G. Downes. Reliability analysis in responsive mission planning for autonomous vehicles. In *Proceedings of the 26th International System Safety Conference*, Vancouver, 2010.

2 citation(s) on 1 page(s): 42.

[6] R. a. La Band and J. D. Andrews. Phased mission modelling using fault tree analysis. *Proceedings of the Institution of Mechanical Engineers, Part E: Journal of Process Mechanical Engineering*, 218(2):83–91, 2004.

2 citation(s) on 2 page(s): 40 and 45.

[7] L. M. Bartlett and J. D. Andrews. Efficient basic event ordering schemes for fault tree analysis. *Quality and Reliability Engineering International*, 15(2):95–101, 1999.
1 citation(s) on 1 page(s): 191.

[8] L. M. Bartlett and J. D. Andrews. An ordering heuristic to develop the binary decision diagram based on structural importance. *Reliability Engineering and System Safety*, 72(1):31–38, 2001.
3 citation(s) on 3 page(s): 106, 108, and 191.

[9] L. M. Bartlett and J. D. Andrews. Comparison of two new approaches to variable ordering for binary decision diagrams. *Quality and Reliability Engineering International*, 17(3):151–158, 2001.
3 citation(s) on 3 page(s): 106, 108, and 191.

[10] M. Bouissou, F. Bruyere, and A. Rauzy. BDD based fault-tree processing: a comparison of variable ordering heuristics. In *Proceedings of European Safety and Reliability Association Conference, ESREL97*, 1997.
4 citation(s) on 2 page(s): 106 and 108.

[11] R. E. Bryant. Gragh-based algorithms for Boolean funtion manipulation. *IEEE Transactions on Computers*, 35:677–691, 1986.
4 citation(s) on 4 page(s): 27, 28, 74, and 106.

[12] G. R. Burdick, J. B. Fussell, D. M. Rasmuson, and J. R. Wilson. Phased mission analysis: a review of new developments and an application. *IEEE Transactions on Reliability*, 26(1):43–49, 1977.
7 citation(s) on 4 page(s): 132, 133, 135, and 172.

[13] Y. Dutuit and A. Rauzy. A linear-time algorithm to find modules of fault trees. *IEEE Transactions on Reliability*, 45(3):422–425, 1996.
1 citation(s) on 1 page(s): 18.

[14] J. D. Esary and W. J. Hayne. Properties of an approximate hazard transform. *Naval Research Logistics Quarterly*, 22(1):41–53, 1975.
3 citation(s) on 2 page(s): 74 and 133.

[15] M. Fujita, H. Fujisawa, and N. Kawato. Evaluation and improvement of Boolean comparison method based on binary decision diagrams. In *Computer-Aided Design,*

1988. ICCAD-88. Digest of Technical Papers., IEEE International Conference on, pages 2–5, 1988.
2 citation(s) on 2 page(s): 106 and 108.

[16] M. Fujita, H. Fujisawa, and Y. Matsunaga. Variable ordering algorithms for ordered binary decision diagrams and their evaluation. *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, 12(1):6–12, 1993.
2 citation(s) on 2 page(s): 106 and 108.

[17] W. S. Jung, S. H. Han, and J. Ha. A fast BDD algorithm for large coherent fault trees analysis. *Reliability Engineering and System Safety*, 83(3):369–374, 2004.
7 citation(s) on 5 page(s): 132, 136, 137, 140, and 172.

[18] W. S. Jung, S. H. Han, and J-E. Yang. Fast BDD truncation method for efficient top event probability calculation. *Nuclear Engineering and Technology*, 40(7):571–580, 2008.
3 citation(s) on 3 page(s): 132, 143, and 145.

[19] T. Kohda, M. Wada, and K. Inoue. A simple method for phased mission ananlysis. *Reliability Engineering and System Safety*, 45(3):299–309, 1994.
2 citation(s) on 2 page(s): 39 and 44.

[20] S. Minato. Zero-suppressed BDDs for set manipulation in combinatorial problems. In *Proceedings of the 30th International Design Automation Conference*, pages 272–277, 1993.
1 citation(s) on 1 page(s): 132.

[21] S. Minato, N. Ishiura, and S. Yajima. Shared binary decision diagram with attributed edges for efficient Boolean function manipulation. *Design Automation Conference, 1990. Proceedings., 27th ACM/IEEE*, pages 52–57, 1990.
2 citation(s) on 2 page(s): 106 and 108.

[22] Y. Mo. Variable ordering to improve BDD analysis of phased-mission systems with multimode failures. *IEEE Transactions on Reliability*, 58(1):53–57, 2009.
2 citation(s) on 2 page(s): 107 and 108.

[23] Y. Mo, J. Han, Z. Zhang, Z. Pan, and F. Zhong. Approximate reliability rvaluation of large-scale distributed systems. *Journal of Information Science & Engineering*, 30(1):15, 2014.

1 citation(s) on 1 page(s): 155.

[24] Y. Mo, F. Zhong, Q. Yang, and L. Zhang. The design and application of fault tree benchmark. In *2010 Fifth International Conference on Internet Computing for Science and Engineering*, pages 64–68, 2010.
2 citation(s) on 2 page(s): 101 and 193.

[25] Y. Mo, F. Zhong, X. Zhao, Q. Yang, and G. Cui. New results to BDD truncation method for efficient top event probability calculation. *Nuclear Engineering and Technology*, 44(7):755–766, oct 2012.
2 citation(s) on 2 page(s): 145 and 146.

[26] M. Modarres, M. Kaminskiy, and V. Krivtsov. *Reliability engineering and risk analysis: a practical guide, second edition.* CRC Press, second edition, 2009.
1 citation(s) on 1 page(s): 21.

[27] I. Mura and A. Bondavalli. Markov regenerative stochastic Petri nets to model and evaluate phased mission systems dependability. *IEEE Transactions on Computers*, (12):1337–1351, 2001.
2 citation(s) on 2 page(s): 37 and 74.

[28] A. Pedar and V. V. S. Sarma. Phased-mission analysis for evaluating the effectiveness of aerospace computing-systems. *IEEE Transactions on Reliability*, 30(5):429–437, 1981.
1 citation(s) on 1 page(s): 12.

[29] J. Poole. *A fast reliability analysis for unmanned aerial vehicles performing a phased mission.* PhD thesis, Loughborough University, 2011.
11 citation(s) on 8 page(s): 39, 42, 51, 56, 58, 69, 71, and 149.

[30] D. R. Prescott, J. D. Andrews, and C.G. Downes. Multi-platform phased mission reliability modelling for mission planning. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability Analysis*, 223(1):27–39, 2009.
3 citation(s) on 3 page(s): 65, 69, and 74.

[31] D. R. Prescott, R. Remenyte-Prescott, S. Reed, J. D. Andrews, and C.G. Downes. A reliability analysis method using binary decision diagrams in phased mission planning. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability Analysis*, 223(2):133–143, 2009.

7 citation(s) on 6 page(s): 42, 45, 70, 74, 119, and 147.

[32] D. M. Rasmuson and N. H. Marshall. FATRAM-A core efficient cut-set algorithm. *IEEE Transactions on Computers*, R-27(4):250–253, 1978.
1 citation(s) on 1 page(s): 24.

[33] A. Rauzy. New algorithms for fault tree analysis. *Reliability Engineering and System Safety*, 40(3):203–211, 1993.
11 citation(s) on 10 page(s): 12, 27, 30, 31, 34, 74, 78, 136, 139, and 142.

[34] R. P. Rechard. Historical relationship between performance assessment for radioactive waste disposal and other types of risk assessment. *Risk Analysis*, 19(5):763–807, 1999.
1 citation(s) on 1 page(s): 16.

[35] S. Reed. *Methods for the efficient measurement of phased mission system reliability and component importance*. PhD thesis, Loughborough university, 2010.
11 citation(s) on 10 page(s): 75, 79, 81, 83, 84, 101, 104, 105, 178, and 188.

[36] S. Reed, J. D. Andrews, and S. J. Dunnett. Improved efficiency in the analysis of phased mission systems with multiple failure mode components. *IEEE Transactions on Reliability*, 60(1):70–79, 2011.
26 citation(s) on 18 page(s): 14, 42, 51, 53, 75, 79, 81, 83, 84, 89, 91, 101, 104, 105, 153, 178, 179, and 188.

[37] R. Remenyte-Prescott. *System failure modelling using binary decision diagrams*. PhD thesis, Loughborough University, 2007.
5 citation(s) on 2 page(s): 106 and 108.

[38] R. Remenyte-Prescott, J. D. Andrews, and P. W. H. Chung. An efficient phased mission reliability analysis for autonomous vehicles. *Reliability Engineering and System Safety*, 95(3):226–235, 2010.
2 citation(s) on 1 page(s): 42.

[39] E. Shaaban, A. Salem, and A. Moniem Wahdan. An interleaving based algorithm for ordering variables in shared BDDS. In *The 14th International Conference on Microelectronics*, pages 256–259. IEEE, 2002.
1 citation(s) on 1 page(s): 120.

[40] M. Smotherman and K. Zemoudeh. A non-homogeneous Markov model for phased-mission reliability analysis. *IEEE Transactions on Computers*, 38(5):585–590, 1989.
2 citation(s) on 2 page(s): 37 and 74.

[41] A. K. Somani, J. A. Ritcey, and S. H. L. Au. Computationally-efficient phased-mission reliability analysis for systems with variable configurations. *IEEE Transactions on Reliability*, 41(4):504–511, 1992.
2 citation(s) on 2 page(s): 37 and 74.

[42] A. K. Somani and K. S. Trivedi. Phased-mission aystems analysis using Boolean algebra methods. In *Performance Evaluation Review: Processsing 1004 ACM SIGMETRICS conference*, volume 22, pages 98–701, 1994.
3 citation(s) on 2 page(s): 40 and 74.

[43] Z. Tang and J. B. Dugan. BDD-based reliability analysis of phased-mission systems with multimode failures. *IEEE Transactions on Reliability*, 55(2):350–360, 2006.
16 citation(s) on 12 page(s): 14, 42, 51, 75, 76, 79, 87, 89, 104, 105, 153, and 188.

[44] D. Xue and X. Wang. A practical approach for phased mission analysis. *Reliability Engineering and System Safety*, 25(4):333–347, 1989.
2 citation(s) on 2 page(s): 37 and 82.

[45] X. Zang, H. Sun, and K. S. Trivedi. A BDD-based algorithm for reliability analysis of phased-mission systems. *IEEE Transactions on Computers*, 48(1):50–60, 1999.
16 citation(s) on 9 page(s): 42, 51, 52, 53, 54, 74, 75, 76, and 107.

[46] H. Ziehms. *Reliability analysis of phased missions.* PhD thesis, Naval Postgraduate School, Washington, USA, 1974.
8 citation(s) on 7 page(s): 37, 38, 39, 133, 134, 136, and 172.

[47] H. Ziehms. Approximations to the reliability of phased missions. *Naval Research Logistics Quarterly*, 25(2):229–242, 1975.
6 citation(s) on 5 page(s): 132, 133, 135, 172, and 189.