CRANFIELD UNIVERSITY

SHAHANI AMAN SHAH

SYSTEM LEVEL AIRBORNE AVIONICS PROGNOSTICS FOR MAINTENANCE, REPAIR AND OVERHAUL

SCHOOL OF ENGINEERING
DEPARTMENT OF AEROSPACE ENGINEERING

PhD
Academic Year: 2008 - 2016

Supervisor: Dr. Huamin Jia
February 2016

CRANFIELD UNIVERSITY

SCHOOL OF ENGINEERING
DEPARTMENT OF AEROSPACE ENGINEERING

PhD

Academic Year 2008 - 2016

SHAHANI AMAN SHAH

SYSTEM LEVEL AIRBORNE AVIONICS PROGNOSTICS FOR MAINTENANCE, REPAIR AND OVERHAUL

Supervisor: Dr. Huamin Jia
February 2016

This thesis is submitted in partial fulfilment of the requirements for the degree of PhD

ABSTRACT

The aim of this study is to propose an alternative approach in prognostics for airborne avionics system in order to enhance maintenance process and aircraft availability. The objectives are to analyse the dependency of avionic systems for fault propagation behaviour degradation, research and develop methods to predict the remaining useful life of avionics Line Replaceable Units (LRU), research and develop methods to evaluate and predict the degradation performances of avionic systems, and lastly to develop software simulation systems to evaluate methods developed.

One of the many stakeholders in the aircraft lifecycle includes the Maintenance, Repair and Overhaul (MRO) industry. The predictable logistics process to some degree as an outcome of IVHM gives benefit to the MRO industry.

In this thesis, a new integrated numerical methodology called ‘System Level Airborne Avionic Prognostics’ or SLAAP is developed; looking at a top level solution in prognostics. Overall, this research consists of two main elements. One is to thoroughly understand and analyse data that could be utilised. Secondly, is to apply the developed methodology using the enhanced prognostic methodology.

Readily available fault tree data is used to analyse the dependencies of each component within the LRUs, and performance were simulated using the linear Markov Model to estimate the time to failure. A hybrid approach prognostics model is then integrated with the prognostics measures that include environmental factors that contribute to the failure of a system, such as temperature. This research attempts to use data that is closest to the data available in the maintenance repair and overhaul industry.

Based on a case study on Enhanced Ground Proximity Warning System (EGPWS), the prognostics methodology developed showed a sufficiently close approximation to the Mean Time Before Failure (MTBF) data supplied by the

Original Equipment Manufacturer (OEM). This validation gives confidence that the proposed methodology will achieve its objectives and it should be further developed for use in the systems design process.

Keywords: Aircraft maintenance, Prognostics in avionics, Enhanced ground proximity warning system

ACKNOWLEDGEMENTS

Alhamdulillah, this thesis is finally done and submitted. First and foremost, I thank Allah.

I owe so much to my dedicated supervisor Dr. Huamin Jia, who has been very tolerant with me, giving me priceless advice and keeping up with my slow progress. Also not forgetting to Professor John Fielding, who has always been there giving valuable advice and insights. Million thanks to both of you.

My special gratitude goes to my three children who have sacrificed so much, Wan Danial Arif, Wan Zaheen Aina and Wan Ahmad Najmi. I am also grateful to my beloved mother, Norehan Mohamed Tahar and my husband, Wan Mohd Sufian for consistently supporting me throughout. To all my in laws especially Izan, families in Malaysia who has been praying hard for my success, thank you. It has been an unforgettable experience. I would not have done it without all of you.

To my boss then and now: My Deputy President (Deputy President (International, Industrial & Institutional Partnership) of UniKL, Prof. Dato' Hj. Ahmad Zahir Hj. Mokhtar, thank you for having faith in me and giving me once in a lifetime opportunity.

To my special friend: Sara Cave, I can never thank you enough. Thank you for listening and providing consistent support. You will always be in heart.

To my friends: Najwa, Kak Sham, Aishah and Feza, I am blessed to have friends like you.

To my sponsors: MARA and UniKL, I thank you for the financial support I have been receiving endlessly.

Finally, this thesis is dedicated to my late father, Aman Shah Zakaria. You would have been really proud. Love you, Abah.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

ACARS	Aircraft Communications Addressing and Reporting System
AI	Artificial Intelligence
ATE	Automated Test Equipment
ATP	Acceptance Test Procedure
BIT	Built in Test
BITE	Built in Test Equipment
CAA	Civil Aviation Authority
CFR	Constant Failure Rate
CND	Cannot Duplicate
EASA	European Aviation Safety Agency
EGPWC	Enhanced Ground Proximity Warning Computer
EGPWS	Enhanced Ground Proximity Warning System
FAR	Federal Aviation Authority
FMECA	Failure Mode Effect and Criticality Analysis
FTI	Flight Test Instrument
GPWS	Ground Proximity Warning System
GUI	Graphical User Interface
HUMS	Health and Usage Monitoring System
IPD	Initial Production Delivery
IVHM	Integrated Vehicle and Health Monitoring
LRU	Line Replaceable Unit
MIL-HDBK	Military Handbook
MRO	Maintenance, Repair and Overhaul
MTBF	Mean Time Between Failure
MTBUR	Mean Time Between Unit Repair
NFF	No Fault Found
NFI	No Fault Indicated
NTF	No Trouble Found
PDF	Probability Distribution Function
PF	Particle Filter
POF	Physics of Failure
RTOK	Retest OK
SAE	Society of Automotive Engineers
T2CAS	Terrain and Traffic Collision Avoidance System
TDU	Terrain Display Unit
TPS	Test Program Sets

CHAPTER 1

INTRODUCTION

This chapter presents an overview of the research, defines the research problems in the area of study and justifies the significance of carrying out this research. Most importantly, this chapter provides the research aim, and also the objectives of this thesis. A brief overview of methodology and thesis outline is also presented at the end of this chapter.

1.1 Diagnostics and prognostics

Commercial airlines today are facing many challenges in maintaining their aging aircraft fleet. Avionics systems are of particular concern due to rising problems with reliability and obsolescence as these components age (Czerwonka, 2000). Aircraft availability will be affected and the maintenance costs can rise dramatically if these problems are not addressed appropriately. Therefore there is a need to provide the aircraft operator with means of identifying critical system or components in advance of the onset of deteriorating performance to allow corrective measures to be taken early which then can prevent unnecessary hardships.

This thesis was conducted to show the need of prognostics towards the avionics equipment of aircraft. By using prognostics techniques for evaluating the avionics equipment health condition user can be get an indication of the equipment’s remaining useful life cycle. This will help the operator to control and plan the maintenance activities, thus improving the efficiency of the aircraft operation and also the operation cost. Implementation of prognostics in aviation field focusing on the avionics equipment will bring a new standard in terms of maintenance activity and increasing the availability and reliability of the aircraft fleet thus also bring a huge savings towards airlines operators.

On-board diagnostics and prognostics are commonly installed in high-value critical assets for the purpose of informing users of the assets' health conditions. The importance of monitoring the health condition is due to an increase in size and nature of fleet with a number of evolutions that has taken place in the aviation industry. A lot of information can be gathered from the 'Prognostics and Health Monitoring (PHM) system. The UK Tornado had a maintenance Data Panel in its design as well as structural and engine health monitoring to identify faults to LRU and LRM in 1972. The EAP designed in 1982 had a computing system extracting data from the data bus and feeding a maintenance data panel which displayed faults in English Language to identify faults to card level. In this way, information was used to identify LRUs to remove from the aircraft (1st line) and then used at the supplier line to identify modules for repair (4th line).

In aircraft, faults can be found in flight by using the Built-in Test (BIT), which is defined as an airborne hardware-software diagnostics tools, recognised to be used as early as 1950s (Pecht et al 2001). BIT however predominantly focuses on diagnostic means to identify and find faults whereas prognostic system will do both diagnostics as well as prognostics. Similar to BIT, the nature and concept of prognostics depends on the parameter of equipment it monitors. Prognostics systems can be designed to assess from the lowest level of component to the highest level of system. A reliable and proper prognostic system must be developed to be able to both provide accuracy and generalisability (Justice et al., 1999).

The main goal of this research is to develop a prognostics methodology known as the 'System Level Airborne Avionic Prognostics' (SLAAP) which aims to predict fault for avionics system that is to be used by the Maintenance, Repair and Overhaul (MRO). Airborne prognostics at system level is intended to allow for deferred maintenance and aircraft is able to be dispatched with known and accepted failure condition. The system is enabled through the use of ACARS and data links where information on identified fault can be passed to the ground crew even before landing. In the modern airliner and private business jets, civil avionics provides vital aspect of navigation, human-machine interface and communication system (Moir et al., 2013).

SLAAP comprises of both diagnostics and prognostics capabilities that gathers fault data from LRUs and help identify problems closer to root-cause. At this level of analysis, SLAAP is believed to provide confident prediction results when uniquely triggered faults are associated with the operating environment when aircraft is in-flight.

Fault diagnostics refers to the process of detecting, isolating, and identifying an impending or incipient failure condition that affect components or systems. There are several events that lead from fault to failure. During the fault diagnostic stage, the system affected can still be operational although it is functioning at a degraded condition. Failure diagnostics on the other hand is detecting, isolating and identifying a system that has ceased operation. Specifically, the term 'fault detection' is used when an abnormal operating component or system is detected and reported while 'fault isolation' is the stage of determining which component or system is failing or has failed. Fault identification is the term for estimating the nature and extent of the fault (Vachtsevanos, 2006).

In general, fault diagnostics is defined as a set of activities to assess the health state of vehicle and its components. From the set of available indications, the diagnostics process determines the root cause in order to explain what has gone wrong. Sensors and crew observation are part of diagnostic process. When a failure occurs, it may not be just from one source of fault. The role of fault diagnostics system is to correctly identify the root cause of the problem. Out of the many methods used for fault diagnosis, these three methods have been most extensively used for fault diagnosis (Aaseng, 2001) : rule-based systems, which usually rely on the 'if-then' analysis; condition-based systems that uses empirical data from past failure and model-based systems deriving failure causes from description of the system components, the relationship between components and information about symptoms related to the components.

A prognostic system can be defined as a process of predicting the failure occurrences of a system by assessing the extent of deviation or degradation from expected normal operating conditions (Pecht, 2008). As soon as fault is detected, actionable decision is made and status of equipment is hoped to be improved immediately. Optimum capability of prognostics approach is in the precision at predicting the failure time or the remaining useful life of a subject. In order to prevent any critical failure, it is important to understand the behaviour of the equipment. (Jie Gu et al., 2007) in a study reported that there were three different methodology of prognostics when dealing with electronics. Namely, using expendable prognostic cells, such as “canaries” and fuses, that fail earlier than the host product to provide advance warning of failure; (2) monitoring and reasoning of parameters, such as shifts in performance parameters, progression of defects, that are precursors to impending failure; and (3) modelling stress and damage in electronics utilizing exposure conditions (e.g., usage, temperature, vibration, radiation) coupled with physics–of–failure (PoF) models to compute accumulated damage and assess remaining life. The simplest form of prognostics is said to be life usage model which is said to be applicable to components that are mass produced (Schwabacher, 2005). Life usage model uses statistics data to calculate the remaining useful life and combine large sample of component to be analysed statistically to analyse usage data.

A study has outlined four fundamental notions for methods in predicting remaining useful life, which are: electromechanical systems age as a function of use, passage of time and environmental condition; component aging and damage accumulation is a monotonic process that involves physical and chemical composition of individual component; signs of aging are detectable prior to failure over time; correlate signs of aging with a model of component aging and thereby estimate remaining useful life of individual components (Saxena et al., 2008).

Prognostics is often associated with condition-based maintenance where; prognostic system decides when maintenance actions are required to be done. This condition-based method is preferred over time-based or event-driven maintenance methods, ideally because it results in less system downtime and only required maintenance actions are taken into consideration.

Similar to diagnostics methodology, prognostics methodologies are categorised into three in the field of complex engineering system; which are physics based model of a system, experts system approach (rule-based) and data-driven (data mining) approach (Schwabacher, 2005). Schwabacher argues that algorithms that use data-driven approach may be the way forward instead of using a hand-built model based because prognosis learns model directly from data. Luo et. Al (2008) on the other hand, describes prognostics methodology in three different approaches which are knowledge based, data-driven and model-based. He developed an integrated prognostic process for an automotive suspension subsystem via model-based simulations. The model-based approach he utilised, describes what Schwabacher refers as physics-based model approach. The models were constructed based on different random load conditions (modes). In the model, an Interacting Multiple Model (IMM) is used to track the hidden damage for deterioration monitoring and the remaining-life prediction was performed by mixing mode-based life predictions via time-averaged mode probabilities. Currently, a vast number of researches have been done in prognostics but there is inadequacy for system level consideration of prognostics researches. Mostly only addresses the prognostics of individual component and subsystems (Amit et al., 2001).

In a paper on the impact and potential benefit of standardisation supporting interoperability of PHM, Sheppard et. al. (2008) highlighted that the focus of prognostics actually lies in area of being able to predict from information about some system state when significant future event affecting the performance of the system such as failure might occur. This estimation comes about to predicting remaining useful life of a component or a system (Sheppard et al., 2008). However, they have suggested that the term time to failure (TTF) as being more appropriate for calculating system level prognosis. The term TTF in their context is a measurement of a system state to some failure or interest in the system as opposed to (Vachtsevanos, 2006) definition of TTF. (Hines and Usynin, 2008) also highlighted that prognostics modules are usually developed to predict one of the following:

1. Remaining Useful Life (RUL), which is the amount of time, in terms of operating hours, cycles, or other measures the component will continue to meets its design specification.
2. Time to Failure (TTF) which defines the time a component is expected o fail
3. Probability of Failure (POF). Which is the probability distribution of failure of the component

Correspondingly, Amit et. al. (2001) have defined a similar classification to prognostics which includes TTF, RUL, POF and the probability that component life, will end before the next maintenance or inspection. They have however, categorise prognostics differently than previous researchers, whereby methods of prognostics is categorised by the type of information the prognosis hold and use. The type of methods is defined as Time-to-failure data-based, Stressed-based and Effects-based. By looking at the kind of information retrieved in Time-to-failure data based approach, it is merely statistical methods as it uses history data and fit them into any distribution function of choice. The stressed based method mentioned is actually the data driven approach as it uses prior observations of explanatory variables and correlate them with time to failure to predict the time to failure of a component. The suggested model to be used in this method is somehow really interesting as this model has not been applied in this field. This technique merges failure time and stress data to modify baseline hazard rate to form new hazard rate. The last method described as Effects-based Prognostics seems so close to what have been categorised as probability based methods according to other researches which uses degradation information to track failure. For this method, Markov Chain-based model was used.

1.2 Research problem

The maintenance, repair and overhaul (MRO) business are in great demand, thus forcing the airlines to depend on third party parts suppliers and services in order to aircraft maintained. Some other problems that the MRO industry are facing relate to the logistics network, owing to the nature of demands for aircraft maintenance repair parts, which airline operators perceive difficulties in parts demand forecasting (Ghobbar and Friend, 2002). On the same basis, MRO network performance is to be agile and lean at the same time (Pipe, 2008). In order to be able to provide the right part at the right time, there is a need to forecast the individual systems or subsystems to predict maintenance time. This is when prognostics approach comes into play. With the implementation of avionic prognostics, significant improvements on current maintenance process with the reduction of No Fault Found (NFF), Retest OK (RTOK) and Can Not Duplicate (CND) incidences will then be provided.

As a result, there will be fewer opportunities to remove a good unit, and a higher probability that any random component removed will be the faulty unit. Thus, leading to the research problem that this research study addresses:

“What kind of prognostics approach is best to be developed to improve maintenance process and availability of avionics systems to be specific and aircraft in general?”

1.3 Research aim and objectives

The aim of this research is to fulfil the objectives of condition-based maintenance which is to optimise availability of high-value critical asset whilst reducing overall maintenance cost through development of prognostics. This methodology is aimed to find the time-to-failure of a system to provide ample time for maintenance personnel to take action before any avionic equipment fails. System in this context refers to the level where prognoses will be analysed. An increase in system complexity and component quality has resulted in a shift from component level towards system level prognostics. This research work involves the integration of three research subjects that are prognostic methodology, degradation model and time-to-failure prediction. Four research objectives were identified to complete the aim of this research.

The objectives are to:

1) analyse the dependency of avionic systems including Line Replaceable Units (LRU) for fault propagation behaviour degradation
2) research and develop methods to predict the remaining useful life of avionics LRUs
3) research and develop methods to evaluate and predict the degradation performances of avionic systems
4) develop software simulation systems to evaluate methods developed above considering aircraft environment and flight conditions in which avionics experience

The **first** objective will focus on the relationship of components in the LRUs and LRMs which affect the fault propagation of the system. Valuable information provided at this stage will help provide precise fault to failure recognition for prognostics for avionics which needed more attention.

The **second** objective uses the first objective's results to develop the prediction of remaining useful life of LRUs/LRMs in the avionic system intended to study.

The **third** objective is then to offer a higher level of prognostic methodology that is to include the degradation behaviour and the performance of the avionic system. Currently, the prognostics methodology is only focused on mechanical parts of aircraft.

The **last** objective is to put everything in a nutshell by deriving a software system which is able to evaluate and simulate aircraft environment and flight condition.

This study carries a great impact in the avionics industry whereby, SLAAP provides an optimised solution for maintenance, repair and overhaul offering new enhanced troubleshooting management.

1.4 The proposed model

Although there is a significant amount of published work on developed methodology for diagnostics, only a handful was reported on prognostics. There are even less studies dedicated specifically to avionics system prognostics at system level. The main contribution of this research work is an improved approach for prognostics in the area of airborne avionics system for the use of the maintenance personnel. The characteristics that distinguish this knowledge structuring schema as an innovative approach are as follows:

- The approach is intended to model design problems according to leading-edge theories and models of design.
- The approach makes a step forward in prognostics by paying attention to the enhancement of condition-based maintenance.

Using the proposed prognostics methodology, the failures of avionics systems are expected to be handled more effectively by delivering real time advisory to secure operators next flight and identify corrective maintenance. On the other hand, using the degradation signatures, current avionics health condition as well as remaining life can be predicted. In addition, a correlation method to validate the confidence level for release of aircraft with environment condition parameter incorporated in the analysis to provide better prognostics results. Model to assess the current health and time to failure is proposed as in Figure 1-1:

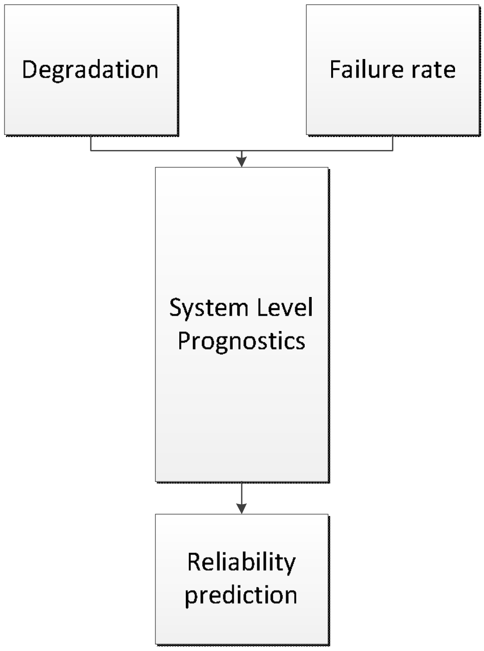

Figure 1-1: Mainframe of methodology used in this study

The proposed research methodology consists of three main branches which are the degradation model, usage of relevant failure data and reliability prediction. The degradation model makes use of Markov modelling techniques while making use of the failure data, and the reliability prediction uses Cox' Regression theory to correlate life degradation (failure) with environmental condition such as temperature. These new integrated models will then be solved to determine the time to failure of the system

1.5 Thesis outline

This thesis is presented in nine chapters as illustrated in Figure 1-2:

Chapter	Title	Synopsis
1	Introduction	Describes the need and motivation of this research and introduces the problem statement and research questions. This section provides an overview of the need for prognostics, current maintenance practices and issues in the area of maintenance, repair and overhaul.
2	Prognostic Techniques for Avionics	Comprehensively reviews the prognostics and its application, also its advantages and drawbacks. This section provides the techniques currently used for prognostics in avionics.
3	Research Methodology	Discusses the methodology proposed and how they merge in this study.
4	Remaining Useful Life (RUL) Prediction Methods in Line Replaceable Units (LRU)	Describes the fundamental theory behind research methodology that was chosen to be used in this research work.
5	Terrain Awareness and Warning System	Describes the purpose, composition, functions, and specification related to TAWS/EGPWS.
6	Results	Embeds the validation using field data into the discussion of the results.
7	Discussion	Discusses the findings that could be potential solutions for industry problems.
8	Conclusion	This chapter concludes the research work and gives direction for further research.

Figure 1-2: Thesis layout

CHAPTER 2

PROGNOSTICS TECHNIQUES FOR AVIONICS

The aim of this chapter is to analyse evolutionary findings and techniques used in handling prognostics problems. Literature studies in pertinent to prognostics, prognostics methodology and prognostics effects in relation to aviation maintenance are studied and analysed for understanding. This chapter also provides a background study on how the problems are tackled by other researchers. In this chapter, prognostic studies will be identified by its classification and in accordance to the current trends and applications. Finally, this chapter will determine the research gap in the area of prognostic methodology for airborne avionics.

2.1 Introduction

In 2001, Federal Aviation Administration (FAA) amended a ruling on the operating rules which requires certain airplanes to be equipped with an FAA-approved terrain awareness and warning system (TAWS) or the enhanced ground proximity warning system (EGPWS). Such equipment was designed to prevent Controlled Flight into Terrain (CFIT). According to a paper by Airbus (2014) entitled "Commercial Aviation Accidents 1958-2013 – A Statistical Analysis", CFIT, which refers to in-flight collision with terrain, water or obstacle without indication of loss of control, CFIT contributes about 23% of total number of accidents since 1994 under the fatal accidents category. A fatal accident in this case is an event in which at least one passenger or crewmember is fatally injured or later dies of his or her injury. Both the TAWS and the EGPWS, like other avionics system are quite a challenge to monitor. This is reflected by a study done for electronics equipment where the wear-out time has been longer than the life cycle of the whole system (Sundström et al., 2008). At times also, electronics equipment fails without any definite measurable signs of fault. Therefore, it is important to establish a proper prognostic method and precursor of failure to be able to detect, isolate, and achieve prognostics outcome. Particularly important is the system level prognostics as compared to the single component level because, when faults or failure occur in airborne avionic system, technicians will simply remove an LRU rather than an electronic component such as a resistor or a memory card in the LRU. System level prognostics referred in this study is the prognostics levels of between level 4 and level 5 as summarised in Table 2-1.

Table 2-1: Failure site and prognostics level in electronics (Jie Gu et al., 2007)

Prognostic level	Site
Level 0	chip and on-chip sites
Level 1	parts and components that cannot be disassembled and reassembled with the expectation that the item would still work
Level 2	circuit board and interconnects connecting the components to the circuit card
Level 3	enclosure, chassis, drawer and connections for circuit cards
Level 4	entire electronic system (LRU/ notebook)
Level 5	multi-electronic systems and external connections between different systems (LRU and cockpit display)

As shown in the Table 2-1, each level is grouped according to similar electronic interaction where group level 0, being the lowest level that describes chips and on-chips sites. Regular electronic components like transistors and resistors alike fall under the grouping of ‘level 1’. Level 5 is the highest level considering the intergroup interaction among multi-electronic systems.

As with the electronics system in the aircraft, the LRUs (level 4) are removed to enable a component (level 2 or 3) to be removed, hence a quicker turn-around time for the aircraft. Many LRUs in Boeing 757/767, Airbus A300/A310

McDonnell Douglas DC-10 and Lockheed L-1011 used to have digital codes to display types of fault that was detected (Vachtsevanos, 2006). Over time, this became less effective as systems became increasingly integrated with each other.

Because of an increase in system complexity and component quality, it is useful to be using the results of a level 1, 2 or 3, as it contributes to the failure of a larger system. However, different prognostic methods are needed to cater each level as complexity accumulates as the levels increase. Intensity of factors affecting degradation may also be of different rates. For example, an increase in temperature at level 0 does increase the rate of degradation at level 5. This though may or may not be affected at the same rate. A hybrid approach to handling this issue maybe the way forward as industries are lacking in hybrid approaches in electronics (Tuchband, 2007).

2.2 Avionics design and development process

The evolution of avionics with an increase in utilisation in avionics technology to be used in engine control and flight control began since 1950s (Moir et al., 2013). The advancement of avionics has been influenced by not only the aerospace industry, military and space but also the modern information technology and communication system existing today.

The improvement of avionics component in terms of trends in integrated circuit development as compared to Moore’s Law can be seen in Figure 2-1. The number of transistors on a chip for microprocessors used in aerospace increased with the advancement in information technology. The effect from this evolution brings not only hardware issues but also software issues where most avionic components rely upon.

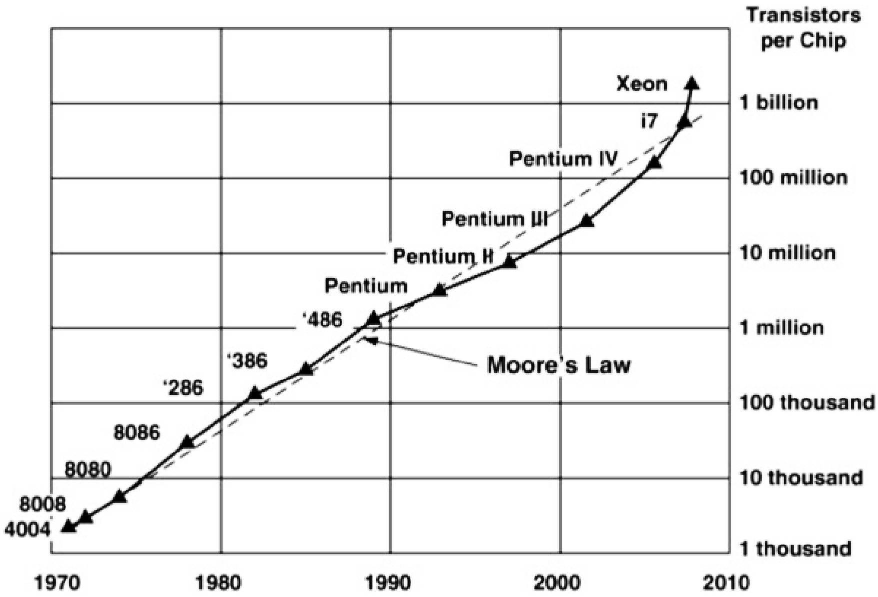

Figure 2-1: Microprocessors used in aerospace application (Moir et al., 2013).

2.3 Importance of prognostics in avionics maintenance process

Maintenance optimization is a process that attempts to find the best balance of the maintenance requirements in terms of contractual, economics, technical, and the resources used to carry out the maintenance program such as workmanship spares, consumables items, equipment, facilities, and others.

When the maintenance optimization is effectively implemented it will improve system availability, reduce overall maintenance cost, improve equipment reliability, and improve system safety. In the former case, optimization is performed to choose the option that generates the largest cost avoidance and or maximizes the availability for an individual system. In the latter, optimization is performed to choose the optimal subsystems to be maintained and meet availability at the enterprise level.

The maintenance efficiency of systems is an important economic and commercial issue. The main difficulties result from the choice of maintenance actions. A bad choice can lead to a maintenance with an over cost that is not acceptable. Because of the increase of involved technologies and the different interactions between components, the decision of a maintenance action is very complex and requires a diagnostic and prognostic analysis. Maintaining such a system basically consists in replacing components that are unable to perform their function by new ones.

Maintenance activities are costly for several reasons. The first one is the issue of aircraft on ground (AOG) during the maintenance phase. The longer the maintenance phase is, the more costly it will become. It follows that the maintenance phase must be reduced to the strict minimal operation, which requires the correct replacement of components. This requires time-on decision relying on an efficient and complete analysis of the health of the system when it is operating.

The second reason causing high maintenance cost is when emergency or sudden failure arises. If a component suddenly fails and the system fully breaks down, it automatically requires some unscheduled maintenance actions which are more costly than scheduled maintenance. To partly avoid this issue, prognostic methods are used in order to perform preventive maintenance. It refers to replacing components during a scheduled maintenance phase that are not faulty yet but that will inevitably become faulty before the date of the next scheduled maintenance phase.

In the maintenance field, the maintenance levels are concerned with grouping the tasks for each location where maintenance activities are performed. The criteria in which the maintenance tasks selected at each level include; task complexity, personnel skill-level requirement, special maintenance equipment and economic measures. In the military, the first level of maintenance process normally starts from where the system is operated and the highest level or the fourth level is usually the OEM. In the commercial aviation industry though, maintenance are categorised by only the line maintenance and the heavy maintenance.

In line with the initial purpose of prognostics which is to reduce costs of operating safely and maintenance efficiency, there are basically three types of maintenance in military or the commercial aviation alike. They are the on-condition maintenance, hard time maintenance and condition-based maintenance. The elaboration on the three types of maintenance is described in the Table 2-2. Unlike the on-condition maintenance and the hard time maintenance, condition-based maintenance is a predictive maintenance process. Luo (2008) described the on-condition and hard time maintenance processes as generally being corrective and preventive. This is because on-condition maintenance process will offer maintenance when needed by monitoring the rate of deterioration of an item, while hard time maintenance is a preventive maintenance strategy where maintenance is done on a timely basis. In comparison to hard time maintenance, the proactive on-condition maintenance measures some condition that is a better predictor of functional failure than time thereby increasing interval between reworks of each unit. That increased interval decreases logistic costs and decreases opportunities for maintenance-induced defects.

The corrective maintenance is also proactive in nature but utilises intelligent sensing and analysing of failure precursors for each item. Corrective maintenance will only be done when breakdown can possibly happen. Further monitoring is needed to ensure parts are replaced or exchanged before they fail. It causes discovery of potential failures rather than allowing functional failures to occur. It localizes the requirements for logistic support by discovering these failures at convenient times and locations.

One advantage of corrective maintenance is that part replacements will only be changed when necessary. One downside to it is parts replacement planning. Preventive maintenance on the other hand, follows a timely scheduled and parts are replaced based on trends reported in equipment log to determine the optimum time for parts exchange. Studies have revealed that both corrective and preventive maintenance are not cost-effective. Last but not least is the condition-based maintenance which is carried out in response to a significant deterioration in an equipment or unit. The time to perform this maintenance action is determined by monitoring the actual state of the system, its performance and other condition parameter. This would mean that the system is in its most efficient state and maintenance would be done when it's cost effective.

For condition-based maintenance in avionics maintenance process, Byington et. al. (2004) developed a modular application allowing information to be accessed via personal data assistant (PDA) for the maintenance crew on ground, building upon open architecture designs and utilising reusable, modular components to enhance diagnosis and reduce ambiguity. The advantage of this system they developed is that the study was able to provide less maintenance hierarchy incorporating interoperable technology testing. Nevertheless, the study stops at diagnosis in which it should have been better if prognostics were included, as the condition-based maintenance is best implemented with the employment of prognostics.

Table 2-2: Approved maintenance process recognised by CAA

Types of maintenance	Definition	Characterisation	Time done
On-condition	A preventive process resulting from inspection or testing of a component to determine service continuation.	Corrective maintenance	Timely basis
Hard time	A preventive process in which deterioration of a component is limited to an acceptable level by maintenance action.	Preventive maintenance	Timely basis
Condition-based maintenance	A process in which information on components are gained from continuously collecting, analysing and interpreting service experience for corrective actions.	Predictive maintenance	When needed

The maintenance, repair and overhaul business are in great demand, resulting airlines to depend on third party parts suppliers and services. So as to keep the costs at the minimum, it is ideal to have a system like the condition-based system that will minimise downtime and have maintenance only when it occurs, while knowing when to act before any fault occurs. Other benefits also include less time spent on inspection and optimised maintenance planning. In many years to come, this kind of maintenance will be practised more frequently as compared to the time based and event driven scheduled maintenance. Usual methods used are model based which needs precise measurement and data, and another method is the data driven. The data driven method can be random that makes it fall under the probabilistic prediction method. This method is considered easier in the sense that it is easier to detail out data. However, with scarce data resources available, this method is useful.

2.3.1 **Degradation or fault occurrence in electronics/ avionic systems**

Electronic components generate generous amount of heat. Components in electronic equipment are stored, packed, and tight to each other, and thus the possibility of overheating can be overwhelming. Whilst this is true, aircraft designers take great pains to provide cooling for avionics equipment by means of air wash, forced air, cooling fans or closed cycle refrigeration system to ensure temperatures do not exceed 70°C through the ambient range of -40°C to +90°C. In fact equipment is usually maintained to function between 20°C to 40°C. Rigorous qualifications testing using DO-160 or MIL-STD-810 is used to gain confidence and evidence to support certification, also includes vibration, shock, and humidity. Deviation from declared condition need to be understood by ground testers so this would be useful flight information to gather. Humidity in tropical climates is often a cause of NFF as a result of tracking on PC boards which disappears when the boards are dried. In this case it is often difficult to treat components in isolation from boards they are mounted on.

It has been reported that as operating temperatures increase, components are prone to failure (Saxena et al., 2008). In effect, they have outlined the four fundamental notions for methods in predicting remaining useful life, which are

- Electromechanical systems age as a function of use, passage of time and environmental condition

- Component aging and damage accumulation is a monotonic process that involves physical and chemical composition of individual component
- Signs of aging are detectable prior to overt failure over time
- It is possible to correlate signs of aging with a model of component aging and thereby estimate remaining useful life of individual components

Unlike mechanical parts such as engines and pumps that are replaced during overhaul if needed, an electronic component is not repaired to ‘as new’ condition. The failure rate of an electronic component changes during its life cycle due to internal and external factors, which create three distinct failure rate zones. The failure rate of a component is relatively constant during its normal operating life, or zone, and then failures are induced by external stresses. Since components are not repaired to ‘as new’, the subcomponents continue to accumulate operating time and eventually begin to fail due to internal stresses. Since internal stresses are added on, the failure rate of a component will increase after its useful life. This period is known as wear out zone.

Electronics degradations are mostly caused by thermal cycling which involve rapid changes in temperature causing thermal expansion and contraction. This has been known to contribute to wire lifts and die solder degradation, chronic temperature and electrical stress, voltage spikes and also by chronic over voltage and over current. In general, electronics wear out are mainly caused by the electromigration, transient electrical stresses, excessive heat, electromagnetic interference, vibration and also mechanical failures. Denson (1998) has stated in his study shown in Figure 2-2 that the majority of causes in electronics are found in parts. The pie chart also shows the other factors contributing to electronics failure in his analysis.

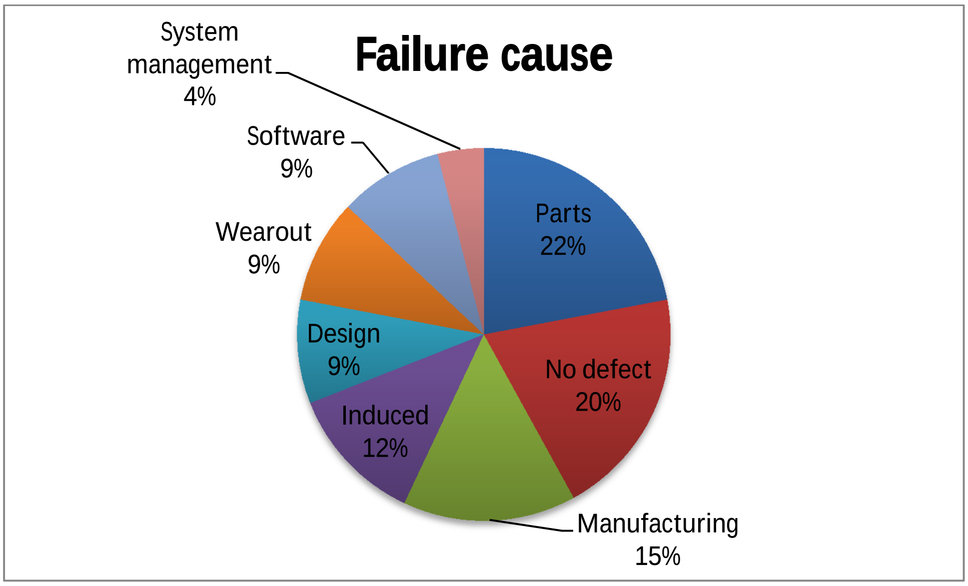

Figure 2-2: Failure causes in electronics (Denson, 1998)

These factors are possible to be monitored through the use of sensors, should there be any prognostic application. To be exact, although it will be quite challenging; it is far worthwhile to apply a prognostic approach in a complex and intensely sensitive equipment, than at a smaller, lower level component..

2.3.2 **No Fault Found (NFF)**

Highly integrated avionics design in newer generation of aircraft puts maintenance personnel in need to be highly skilled in performing their tasks to get aircraft flying again. This scenario is due to the fact that the duration of performing maintenance is associated with man-hour expenditure and aircraft downtime. With aircraft downtime being a problem, which then contributes to the performance evaluation of airline and civil aircraft (Knotts, 1999).

One of the factors contributing to the underlying problem is unable to correctly diagnose problems from reports provided by pilot and data readily available from the aircraft will indirectly increase operating costs for airlines. These problems encountered by flight crews can be challenging for the crew on ground to detect. Sources of fault in equipment are hardly detected (cannot be reproduced) because they only occur intermittently. Sometimes, the problems encountered were poorly described or not properly addressed in the maintenance log. When the problem reported cannot be proven to be faulty, it is called Can Not Duplicate (CND), but when there are no problems found in the findings, it is called No Fault Found.

Direct maintenance costs are contributed by 18% of avionics and electrical unscheduled maintenance with 40% of related equipment removals are classified as No Fault Found (NFF), which means that an LRU will show that condition is faulty while aircraft is off the ground but seems fine when on ground (Wu et al., 2004). Knotts (1999) has cited that an average of 8000 component removals fleet per month in an audit that has taken place at British Airways, whereby a total of 14% of components, across all workshops, were found to have NFF. Certain avionic equipment experienced 30% NFF. Various terms such as Retest OK (RTOK), no fault indicated (NFI), and no trouble found (NTF), are also referring to the inability to replicate field failures during ground run. It has also been found that more than 85% of CND failure in the avionic field will account for more than 90% of total maintenance costs (Williams, et. al, 1998).

The number of documented CND, NFF, and RTOK indicates a large amount of money and manpower spent in the pursuit of high availability and reliability of electronic systems of aircraft (Byington et al., 2004). Some NFF conditions are caused by intermittent faults. Intermittent faults seldom appear unless a unit is in a stressful operating environment. The lack of fault traceable data, such as operating time to failure and environmental conditions when a fault occurred, obstructs the potential ability for effective avionic prognostics and failure predictions on an aircraft. Although Built-In Test (BIT) that was a simple push button that illuminates different colour lights to test for functionality (Bird et al., 2005) has been around for quite a while, it can misidentify faults. Even with the sophistication of Built-In Test equipment (BITE) and Centralised Maintenance Systems, fault detection is still considerably high (Johnson, 1996). With this said, it is possible that the faults were actually generated from the BITE. One incident reported by Johnson (1996) on Lufthansa’s A320 fleet of operation where out of an average 17 LRUs were removed each day, only two were confirmed faulty. Airline operators are constantly faced with irregular operational problems that are developed from unexpected aircraft system failures, which may be followed by a reschedule of flight service or aircraft reroute. Events such as flight cancellations, delay and reshuffling of aircraft maintenance scheduling may also take place (Ghobbar and Friend, 2002). In Malaysia, according to a data taken from Harun, aircraft maintenance costs for Malaysia Airlines in the year 1996/1997 was estimated at USD140 million and aircraft technical delay costs USD5.5 million a year (Harun, 1998).

Therefore, by providing an early warning of failure, enough time before it eventually happens will help plan and organise replacement parts. Prognostics in airborne avionic system is all about providing accurate enough fault data that contingency plan can be scheduled rather than leaving it until breakdown takes place and handle problem as it happens. So, by providing consistent health assessment on aircraft system, NFF can be reduced as fault prediction is done much earlier.

2.3.3 **Flight delays**

One of the problems faced these days are delays. Figure 2-3 illustrates on departure delays causes in the year 2009. Presented in the Table 2-3, are the factors contributing to the delay under the ‘airline’ category. In brief, airlines must foresee defects problem seriously to cater for the public demand which is escalating in the near future. Hence, the relevancy of the prognostic methodology for airborne avionic systems is considered imperative. This is because flight planning and airport planning relate closely to time. As shown in Figure 2-3, delays are mostly caused by airlines, and problems are mainly caused by aircraft defect. Therefore, once prognostics systems are in place, defective components are hoped to be solved just enough time before any planned take-off, thus reducing fault rectification time. It is when unscheduled maintenance is urgently needed that aircraft needs to be on-ground thus causing delays.

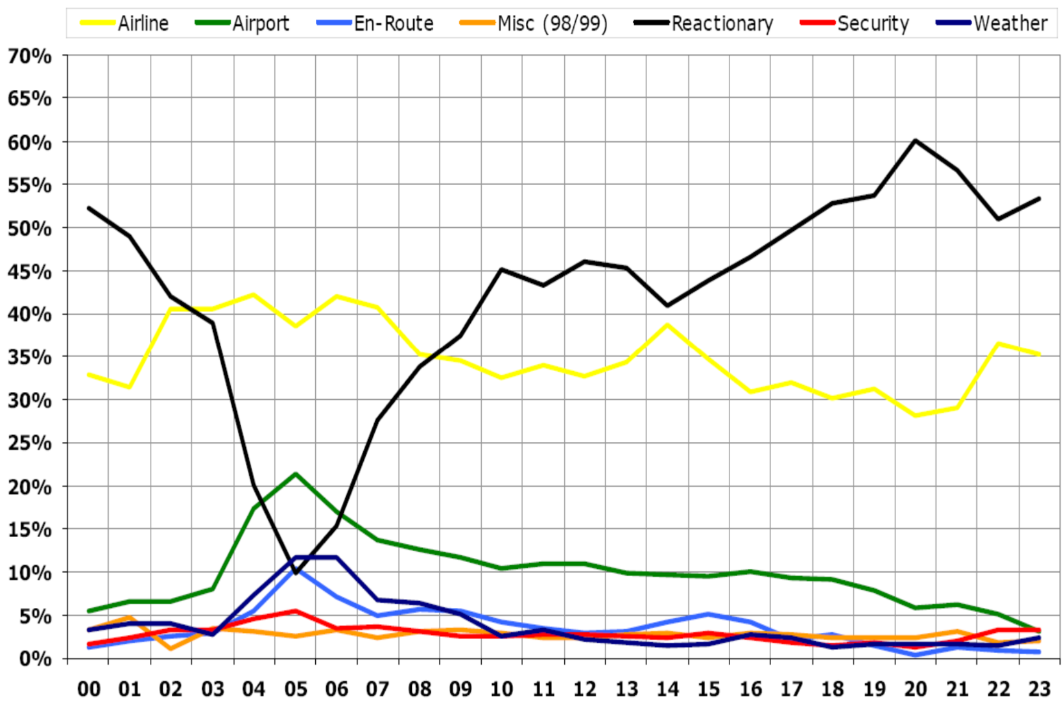

Figure 2-3: Breakdown of departure delay cause by hour (Eurocontrol, 2009)

Table 2-3: Breakdown of technical and aircraft equipment IATA (Eurocontrol, 2009)

Category of Technical & Aircraft Equipment IATA category	Apr-09
41-Aircraft Defects	277,959
42-Scheduled Maintenance	24,815
43-Non-Scheduled Maintenance	49,514
44-Spares and Maintenance Equipment	12,631
45-AOG Spares	3,839
46-Aircraft Change	161,248
47-Stand-By Aircraft	15,862
48-Scheduled Cabin Configuration	2,526

Every change in schedule that is caused by delays will affect costs. Therefore it is the aim of this research to provide a solution to reduce the time taken to isolate the fault or it may lead to deferred maintenance. The total overall time will consequently affect logistics through a well-planned maintenance system.

2.4 Prognostics’ advantages and drawbacks

Prognostics advantages outweigh its drawbacks in certain areas of applications and vice versa in others. The application may be worthwhile when the return of investment is positive. Otherwise, it will only be a waste of time. Some of the benefits of prognostics method are failure avoidance, positive logistics support, maximum life usage, opportunistic maintenance support, fast turnaround time, low aircraft on ground records, better mission planning and most importantly no unexpected breakdowns. Thus, RTOK and CND numbers could effectively be reduced. Historically, machines have relying upon experience of handler or operator. Trouble arises when the operator is off site. Machines are only looked at when they are no longer function. No warning or signs provided. With prognostics, complex machines can provide early warning and ability to exploit useful data effectively. Predictive prognostics is indeed a need and theoretically promising.

Studies have reasoned that when it is possible to use available data to perform diagnostics, it is not possible to stretch on further to detect and monitor degradation. Of course the benefits will show in a cycle as everything will boil down to costs. With application of prognostics, machines’ failure can be predicted, thus logistics and maintenance support will be well planned. This then reduces impromptu downtime and unwanted surprises. Because actions can be taken through known information (data), better scheduling could be realised. Organised actions and precise decisions help in controlling costs within budget, which is the ultimatum of any companies.

Some challenges of prognostics on the other hand involve the preparation and prognostic installation costs. A well-planned study must be thoroughly laid out prior to the implementation. Not all areas need prognostics works. Something simple and cheap does not need anything complicated. For example, when a light bulb fails, it just fails. A sensor would help monitor current flow but monitoring a light bulb is not critical. The application of prognostic is worthless as light bulbs are cheap and are easily fixed. However, it is not something very impossible as the capability of diagnostics and prognostics are desirable.

As a conclusion, prognostics has been shown to be beneficial for health management of systems, and provides a number of potential benefits including, methods to assess the reliability of a product in its actual life cycle conditions to determine the advent of failure and mitigate system risks, ability of a service or a system to be functional when it is requested for use or operation and cost avoidance systems and can be summarized as:

a) Avoiding of unexpected failures with consequences reduction of unscheduled maintenance actions:
    - Minimizing the cost of unscheduled maintenance
    - Increasing availability
    - Reducing risk of loss of system
    - Increased human safety
b) Reduction in no-fault-founds:
    - The data collected and continuous monitoring used in prognostics can be helpful to flight line and shop maintenance personnel in locating a faulty item. The RTOK problem is well known and has resisted many attempts at reducing it. The accumulated damage information provided by prognostics assists in localizing a problem and informs the test more likely to reveal it.
c) Minimizing loss of remaining life:
    - Minimizing the amount of remaining life thrown away by scheduled maintenance actions
d) Reduction in the required number of repair stations and stores locations:
    - The ability to control the occurrence of maintenance actions leads to the ability to control the location at which they occur, thus reducing the required number.
    - In addition, foreknowledge of spares requirements allows them to be delivered ‘just-in-time’, thus reducing the spares stockholding levels. This leads to a substantial simplification of the spares supply chain.
e) Improved repair:
    - Better diagnosis and fault isolation

- Reduction in collateral damage during repair
- The avoidance of costs associated with unscheduled repairs, such as assets and crew down time, special spares shipments, and replacement crews can be a major cost saving as well as reducing the number of non-mission capable assets.

f) Ability to adjust assets usage according to its actual readiness:

- Presently mission planners have no knowing which of their available assets is most like able to complete the planned mission.

2.5 Cost benefit of prognostics

In order to show the best value direction in implementing prognostic methodology, a cost benefit analysis has been illustrated as described by Janasak and Beshears (2007). The Table 2-4 illustrates the benefits, significantly on the costs, with the implementation of condition-based maintenance. The example shows that the change in the maintenance interval affects the life cycle cost, availability and the percentage of failures avoided. While it is obvious that condition-based maintenance provides greater cost benefits in terms of cost, additional elements may need to be considered and weighed in for optimum results. Some factors that could be taken into consideration include economic cost, mission and safety implications towards implementing each maintenance approach.

Table 2-4: Example of cost benefit study to determine what prognostic feature is available (Janasak and Beshears, 2007)

Sustainment Approach	Unscheduled	Scheduled	Condition-Based
Maintenance Interval/Prognostic Distance	0 hour	1920 hours	396 hours
Mean LOC	$83,319	$181,094	$109,358
Standard deviation LOC	$16,066	$12,505	$21,067
Mean availability	97.47%	95.26%	97.26%
Standard deviation availability	0.61%	0.49%	0.66%
Failures avoided	0.00%	75.52%	64.53%

2.6 Prognostic applications

2.6.1 **Prognostic applications in aerospace platforms**

The Table 2-5 is a survey done by Ofsthun on subsystems in order to elaborate its usages and features normally used in aerospace contexts. In the table, for application of IVHM in avionics, only diagnostics was declared. This was probably because prognostics was quite new then. He has also pointed out that the traditional built-in-tests generally have not provided the accuracy needed to impact the operational efficiency in maintenance. Thus, the overall goal achievement of IVHM should be to have an improved and extension to BIT approaches in subsystems such as avionics. In his article also, Ofsthun highlighted lesson learnt relating specific IVHM users goal to diagnostics and prognostics. This study sees similar needs which include:

- To ensure effective IVHM outcome, prognostics must cover an integrated degradation analysis that can measure equipment performance
- Benefit analysis as well as cost efficiency for maintenance repair and overhaul should be taken into consideration

- A top-level system framework is needed to integrate across subsystems

Wide-spread adoption of integrated health management has been slow due to competing factors that have to be satisfied within the prognostics community. Some issues include the life expectancy of an aircraft and cost versus benefit factor. From an engineering perspective, the development of prognostics to mitigate the greatest risks is dependent upon accurate data collection. The data needed for maturation analysis is usually difficult both to obtain as well as to collect. On the cost-benefit challenges of prognostics, it is best to apply prognostics in areas that are historically the least reliable, have failure modes that greatly impact operation success and comprises of subsystems that are difficult to diagnose.

Table 2-5: IVHM features and techniques used in aerospace platforms (Ofsthun, 2002)

Subsystem type	IVHM features	IVHM techniques
Avionics	Diagnostics	Multilevel false alarm filters Field loadable software/ data modules Vehicle level BIT context correlation
Electrical	Diagnostics	Vehicle electrical supply and distribution status correlated with subsystem failure indications
Actuators	Diagnostics, prognostics	Motor current, temperature, vibration and position sensors compared to a performance model to identify failures and performance degradation
Environmental control	Diagnostics, prognostics	Temperature, pressure, flow rate, vibration and valve position sensors compared to a performance model to identify failures and degradations
Propulsion	Diagnostics, prognostics	Engine monitored for foreign object intrusion and dynamic engine performance parameters compared to a performance model to identify failures and degradations, debris density, particle size measurement in oil, low cycle fatigue, rain-flow analysis and blade temperature
Hydraulics	Diagnostics, prognostics, inspections	Fluid levels, pressures, valve positions monitored to detect leaks identify performance degradations and eliminate manual inspections, debris density, particle size measurement in oil and fluid
Structures	Prognostics, Inspections	Real time intelligent load monitoring using flight control data to minimize scheduled inspections and maximize useful vehicle life, loads, corrosion, implications on composites, load test, strain and pressure.
All	Anomaly detection	Aggregate air vehicle parameters correlated to identify anomalous behaviour requiring further investigation or maturation

Health inspection and monitoring spacecraft and aircraft systems are often difficult and costly, often because relevant sensors cannot be installed at the right places. Therefore prognostics methods have been developed and incorporated in the health management systems of the latest military aircraft and civilian aircraft, in order to reduce the overall life-cycle cost and improve flight readiness. Figure 2-4 below shows a schematic of the PHM process inputs, computations and outputs. For clarity, the figure shows only three sensor inputs and three models. The actual number of these will be much larger in practice.

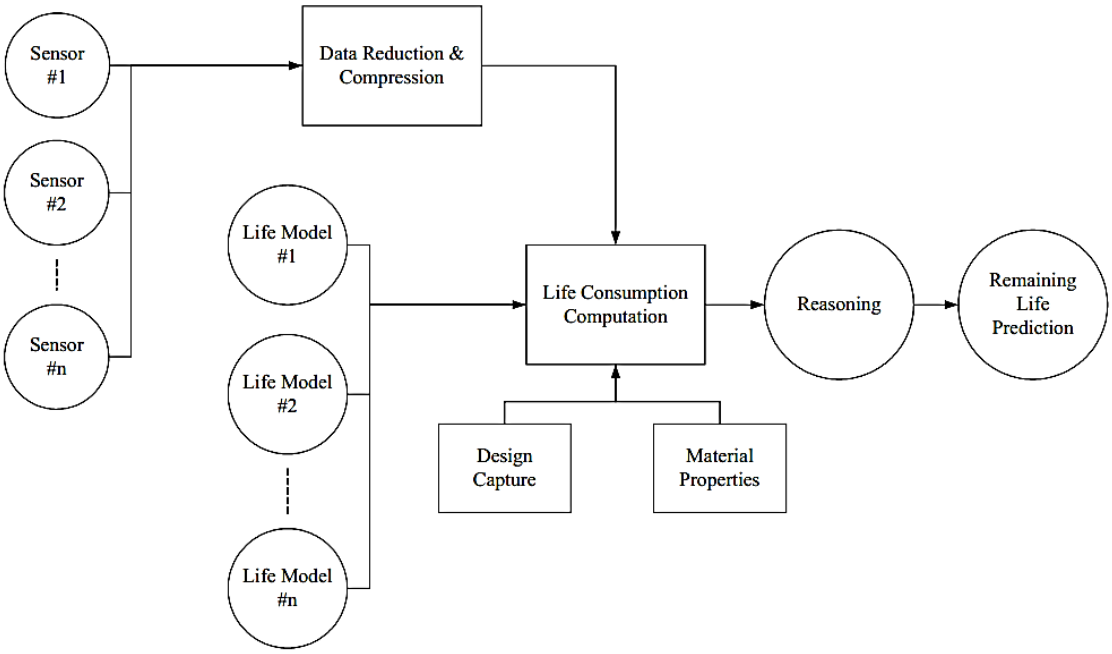

Figure 2-4: Remaining Life Prediction (Wilkinson et al. 2004)

Sensor data provided by a variety of on-engine and on-aircraft sensors is first compressed and reduced. This process greatly reduces an otherwise unmanageable quantity of data, without sacrificing a significant amount of information. A life consumption algorithm, utilizing a variety of life models, knowledge of the design ant the constituent material properties, then computes the amount of life used with an associated confidence interval for each of the possible failures sites.

Since there are sources of uncertainty, such as material properties and the physical dimensions of the various structures making up the electronics assembly, there will be corresponding uncertainty associated with the life consumption computation. The reasoning process combines these uncertainties using a decision support system to give a remaining life prediction for the complete system, again with an associated confidence interval.

2.6.2 **Prognostics work in electronic components**

To do prognostics work, there is a need to identify a measureable precursor to degradation. Using the pattern or trend of degradation and statistics of failure for that component, a benchmark can be created to guide the analyses. For better approximation, probability theory is added on to calculate the remaining useful life for the component under study. A relation between prognostic level and failure rate indicates the relevancy of applying prognostics work, which means prognostics are encouraged to be done for higher level of electronics with high failure rate. After identifying the probability theory to be used for this prognostic methodology, the next step is to identify measurable precursor to failure without knowing the physics of failure of the component chosen. As such, looking at the component as a black box, which means analysing the component overall is a solution. True enough, without knowing every detail of parts and components in a system, predicting its life time is not an easy task. Previous researchers have used particle filter algorithm, and many have used artificial intelligence (AI) algorithm but not many have succeeded in showing they have successfully applied to avionic systems as a whole in order to predict its remaining useful life.

2.6.3 **Prognostics in avionics**

Avionics systems combine physical processes, computational hardware, and software systems, and present unique challenges to performing root cause analysis when faults occur, and also for establishing the effects of faults on overall system behaviour and performance. However, systematic analysis of these conditions is very important for analysis of safety and also to avoid catastrophic failures in navigation systems.

This drives the need for integrated prognostics and health management technologies for flight-critical avionics equipment. Flight and ground crews require accurate health state estimates of these critical avionics components, including accurate detection of faults and prediction of time to the functional failure of the avionics system. An understanding of how components degrade is needed as well as the capability to anticipate failures and predict the remaining useful life of electronic components.

Studying and analysing the degradation of these systems in example degradation in performance to improve aircraft reliability, assure in-flight performance, and reduce maintenance costs, therefore it is absolutely necessary to provide system health awareness for electronics systems. In addition to this, an understanding of the behaviour of deteriorated components is needed as well as the capability to anticipate failures and predict the remaining life of electronics systems.

Some of earlier efforts in diagnostic health monitoring of electronic systems and subsystems involved the use of a built-in test (BIT), defined as an on-board hardware software diagnostic tests to identify and locate faults. Studies conducted by on the use of BITs for fault identification and diagnostics showed that they can be prone to false alarms and may result in unnecessary costly replacement, re-qualification, delayed shipping, and loss of system availability.

The persistence of such issues over the years is perhaps because the use of BIT has been restricted to low-volume systems. In general, BITs generally have not been designed to provide prognostics or remaining useful life due to accumulated damage or progression of faults. Rather, it has served primarily as a diagnostic tool.

According to Lou et al. (LOU et al., 2009) airborne equipment failures are divided into two kinds, which are mechanical fatigue and chemical failure. These two kinds are closely affected by the environment where equipment is installed. However, for most aircraft platforms the precise and individual parameters such as temperature or humidity issues are normally intermittent. NFF could happen as a result of tracking boards which disappears when the boards are dried up.

Last but not least, the miniaturization and complexity of electronic integrated circuits (IC) nowadays has remarkably challenged the reliability of technicians to assess the degradation of electronics from the initial beginning of the design process. Even though the integration of circuits has led to the development of precise and accurate techniques for reliability estimation, limited information is available for predicting the entire health over a wide range of environmental and operational life cycle conditions.

2.7 Forecasting methodology

The fundamentals in forecasting methodology are models and methods. Models can be described as mathematical representations of reality and are usually approximate rather than exact. Models are designed to describe the overall framework used to portray reality using mathematical functions. Methods on the other hand are rules or formulas for computing predictions from observed data. So, methodology is not a model, but can be based on a model. This study is based on the approach of developing a method based on a model that represents the failure trend of an avionic airborne equipment system. In forecasting, the types of execution are divided in three classifications, which are: the ‘subjective, univariate or multivariate,’ the ‘automatic or non-automatic’ and the ‘qualitative or the quantitative’. In the first technique of classification (subjective, univariate or multivariate), subjective technique uses judgement, intuition, commercial knowledge or any other relevant information in order to forecast. It is largely based on educated guesses. These can sometimes depend upon past data if available. Normally, these techniques are relatively hard to reproduce. It is because; it is very unlikely that the data is shown explicitly how it is embedded in the system. Univariate on the other hand forecasts by fitting a given variable based on a model of past observations of the given time series. Time series are data that is represented in an orderly pattern or sequence. For example, extrapolation of trend curves, exponential smoothing, the Holt-Winters forecasting procedure, the Box-Jenkins procedure and stepwise auto regression. In this technique, the function form and coefficient are not known and thus needed to be determined. Ordinarily, it can be obtained from historical data. Lastly, multivariate technique is a technique of forecasting a given variable on values of one or more series called explanatory variables. This method is sometimes called causal models. Some examples of this technique are multiple regressions and econometric models.

Forecasting methods can also be classified according to automatic versus non-automatic approach. This is similar to open and closed loop in control systems. Non-automatic refers to an open system and automatic depends on feedback, similar to a closed loop system. Automatic type of forecasting method does not use any human intervention while the other (non-automatic) applies to subjective intervention. Surely, the automatic system gives an added advantage as output can be updated in real-time. But it all depends upon, many factors such as how forecast is to be used, type of time series involved, number of samples observed, duration of forecasting period, skills and experience of observer, and others.

Lastly, forecasting can be divided into qualitative (subjective) and quantitative (Abraham and Ledolter, 2009). Normally, quantitative methods are often given priority and placed in greater reliance than qualitative method although they cannot be domineering or allowed to be the dominating technique. Qualitative data is also perceived as lagged behind in some application in the past years. However, with the development of state of the art tools and software, qualitative methods include simplified indicators so that they are accepted more widely. Qualitative is subjective in nature and is based on intuition. It may or may not depend on past data. Although it is a non-rigorous approach, it is appropriate and reasonable method for some application. Unlike qualitative, quantitative is based on mathematical model or statistical model. It can be reproduced by any forecaster and is suitable for word problems needing numerical representation. The advantage of qualitative method is that it can track mutual influences by putting numbers to a particular statement or forecast. Besides, it can represent real time monitoring when it is applied in dynamic models. One other advantage is that, using quantitative method, it is possible to manipulate information consistently, and in a reproducible manner. This can be done through figures, combining figures, and also by examining and comparing data. With numbers, it allows for greater precision as compared to merely analysis of increasing and decreasing relationship. However, its application is limited in some areas whereby, not all factors can be represented numerically but can be done in matrices or rubric. Quantitative techniques can be classified by deterministic or probabilistic. Deterministic models the relationship between the variable of interest, Y and explanatory or predictor variable, X1, X2, X3. This way, the outcome is exactly determined. These models only assume a constant failure rate but breakdown quickly when the systems go into actual service. It happens on the account of multimode environmental forces that brings about part failures. Part failure can also occur due to the fact that stress management during handling and assembly is not always practiced in a consistent manner.

Deterministic forecast can be made perfect with the skills of interpreting to the degree of forecast models and how good these models are at estimating. This will also depend on the precision and accuracy of the observation done at the initial stage to produce the model. In deterministic forecasts, through observations, diagnosis is presented. Next, appropriate model is applied and lastly, prognosis is formed. It is based on the logic concept of 'if and only if'. Probabilistic (stochastic) method measures movement from the present state to the future state. It is a technique which relies on different methods to achieve an event with a given weightage of probability. Instead of giving definitive information on the magnitude of event occurrence, probabilistic technique uses uncertainty of prediction based on frequency or pattern of event occurrence.

In application to this study, a useful and acceptable way forward is to understand how the parts fail and then determine how one can prevent that from happening prematurely or in a dangerous manner, and establish what can be considered to be a useful working lifespan. It is normal to consider uncertainty in forecasting since no one estimation is definite, for sure. Because forecasting, estimation, prediction and prognosis relate to events happening in the future, it must be presented such that the results establish an 'educated guess' and not simply a wild guess. Thus, in bringing the methodology together, accumulation of ensemble of forecasts with clear and precise model needs to be compiled before value of the subjective probability estimates can be adhered. Basically, when forecasting methodology topics are discussed the main goal is to develop method subjectively in producing an objective outcome with a precise and accurate model in the background.

With few avionic systems that last ten to 20 years in service without difficulty, it is important to know that most aircraft operate for a period far longer than that. The period of useful life for different components and subsystems can vary significantly and the period when the hazard rate is increasing can be difficult to pinpoint. Hardware in this phase of life may have intermittent and differing causes of failure that are hard to isolate and wear out mechanisms can be complex and may exhibit different failure modes. This condition may account for some avionic units sometimes called "a rogue unit."

2.8 Emerging prognostics approaches

Proper methodology must be chosen for suitable application for it will affect the effectiveness of analysis and study projected. Prognostic methodology remains a critical yet unknown area to major areas of research. Proper methodology must be chosen for suitable application for it will affect the effectiveness of analysis and study projected. Another aspect to optimisation of prognostics is analysing how the system functions and how it fails. A way to model this is by understanding functional behaviour and operation of the system. All in all, when discussing about prognostics, either material degradation or functional deterioration which affect system operation is important. For avionics, prognostics focuses on the functional deterioration of system and should be able to predict one of the following:

- Remaining Useful Life (RUL): the amount of time, in terms of operating hours, cycles, or other measures, the component will continue to meets its design specification.
- Time of Failure (TOF): the time a component is expected to fail (no longer meet its design specifications).

- Probability of Failure (POF): the failure probability distribution of the component.

This is largely due to the fact that there are limitations to predicting the future. Predicting, or sometimes referred as forecasting study is an important activity in our daily life. Forecasting methods can be divided into many categories, but generally it can be either point forecast or interval forecast. In this study of electronic airborne equipment system, it is the aim of the study to develop a method to estimate the remaining useful life of the system. Coble used parameter features such as trendability, monotonicity, and prognosability (2010). These features are used to determine the most useful method for individual prognostics case. He classified three categories of methodology which are reliability-based, stressor-based and degradation-based. Reliability-based considers merely historical time to failure data, stressor-based takes the environment condition into consideration, and degradation-based monitors how specific a component reacts for its specific usage.

Because prognostic methodology is related to reliability, it is therefore significant to identify the reliability prediction for application of specified field. Specific application uses different procedural method for their prediction method in the reliability analysis. This is because the society or association in the field has produced a standard, common ground in order to set the benchmark for reliability for each field of study. For example, the military uses Military Handbook 217 (MIL-HDBK 217), as a mechanism for estimating probability of failure for electronics, whereas the automotive industries use the Society of Automotive Engineers (SAE) reliability prediction method for their reliability studies. Table 2-6 lists several reliability prediction methods for different application.

Table 2-6: Reliability prediction method in variety of application

Procedural Method	Application
MIL-HDBK-217	Military
Telcordia SR-322	Telecom
RDF-93 and 2000	Civil equipment
SAE reliability prediction method	Automotive
Siemens SN29500	Siemens products
BT-HRD-5	Telecom
PRISM	Aeronautical & military
FIDES	Aeronautical & military

Typically, failure prediction methods are examined through a mathematical model, so that the state of equipment can be predicted using some series of historical information. Prognostic methodology is generally divided into four main sections, which are model-based, statistical, data driven and probability based. Figure 2-5, shows the generic prognostic method that is currently used in many different practices.

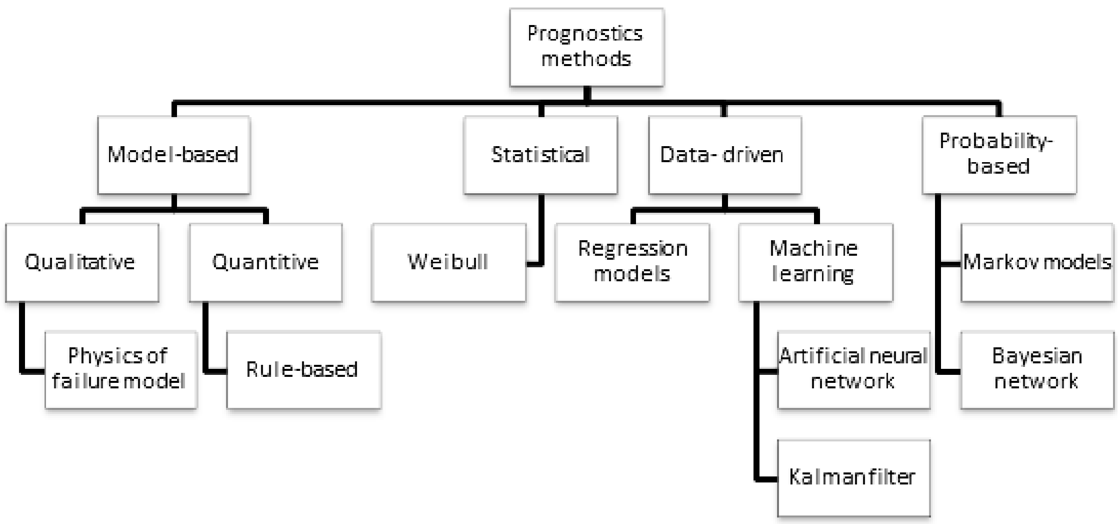

Figure 2-5: Prognostic methods at a glance

2.8.1 **Model-based methodology**

Referring to 'Mathematical Formulation of model based methods for diagnostics and prognostics' by (Jaw and Wang, 2006) the model based approach is more favourable since it provides a more accurate estimation and that it can link naturally to the physics of failure. Particle filter (PF) has been widely applied by many researchers as a method for failure prognostics (Orchard and Vachtsevanos, 2007). They have implemented PF for finding time-to-failure estimation for crack growth analysis. Some of the advantages listed for these particular methods are that compared to classical Monte-Carlo method, sequential importance sampling enables PF to reduce number of samples required to approximate the distribution and at the same time, makes the process faster and more computational efficient. Besides, PF allows information from different sources of measurement to be combined together, systematically. For example, Abbas et al. (2007) have used the same approach in identifying the underlying conditional state probability and use the Particle Filter estimation methods for prediction and filtering. As described in his paper, the underlying methodology is the approximation of the conditional state probability distribution by a swarm of points referred as particles containing a set of weights representing discrete probability masses. Particles can be easily generated and recursively updated given a nonlinear process model and a set of available measurements and an initial estimation for the state pdf, as below:

$$x_k = f_k(x_{k-1}, \omega_k) \longleftrightarrow p(x_k \mid x_{k-1}) \qquad \textbf{(Equation 2-1)}$$

$$z_k = h_k(x_k, v_k) \longleftrightarrow p(z_k \mid x_k) \qquad \textbf{(Equation 2-2)}$$

where x_k is the state of the fault dimension, the parameters involved that are represented by ω_k and v_k are the noise, and f_k and g_k are non-linear functions. The method was applied to the problem of battery grid corrosion where algorithm developed was used to determine the probability of time-to-failure. Particle filter model was used to predict time evolution of fault condition-based on typical automobile pattern.

2.8.1.1 Advantages and drawbacks

The main advantages of using model-based method are that it can detect unanticipated faults and it is highly accurate provided that enough useful features are extracted. Because this method models the true system even with few data, it can produce high reliability results. However, it tends to be computationally prohibitive when applied at system level. This is because one needs to fully understand interaction and dependencies of system in order to build the model correctly. As white noise is propagated at each level, this method tends to also produce large sum of error. Some product usage profile is often predictable but is not always reliable. This method also relies on continuous physical model of a component.

2.8.2 **Statistical methodology**

Statistical-based method is the simplest and useful method, provided that a large history data is available. It is useful where component prognostic models are not warranted due to low level of criticality or low failure occurrence rates (Roemer et al., 2006). Although simple, it can be valuable for maintenance scheduling for electrical or airframe components that have very few sensed parameters. In this case, it is not critical enough to go through the process of developing a physics- based model. Harun (Harun, 1998) have used statistical analysis in determining failure rates for confirmed and unconfirmed removals of parts for Malaysia Airlines System (MAS). The work done was to identify the most efficient time for maintaining a component. Another example where this method is most effective is when failure rate data is easily accessible and can be correlated with specific profile usage, which is predicted to have effects on the failure.

2.8.2.1 Advantages and drawbacks

Using this method, its advantage is that it is workable with small sample size, thus allowing a cost effective component testing. It can also provide useful and simple graphs. Even in the latest technologies, failure mechanism is represented more using Weibull (extensively used in many aeronautical applications). Weibull distribution is widely used in life data analysis due to its versatility and relative simplicity. The most general expression of the Weibull *pdf* is given by the three-parameter Weibull distribution expression:

$$f(T) = \frac{\beta}{\eta}\left(\frac{T-\gamma}{\eta}\right)^{\beta-1} e^{-\left(\frac{T-\gamma}{\eta}\right)^{\beta}}$$ **(Equation 2-3)**

Where:

$$f(T) \geq 0,\ T \geq 0 \text{ or } \gamma,\ \beta > 0,\ \eta > 0,\ -\infty < \gamma < \infty$$

and:

- β is the shape parameter, also known as the Weibull slope
- η is the scale parameter
- γ is the location parameter

Depending on the values of the parameters, the Weibull distribution can be used to model a variety of life behaviours. An important aspect of the Weibull distribution is how the values of the shape parameter, β, and the scale parameter, η, affect such distribution characteristics as the shape of the *pdf* curve, the reliability and the failure rate.

This also proves to be more representative for future LRU. The inadequacy in using this method is that most data are provided with the restrictive assumption of a constant hazard rate function.

2.8.3 **Data-driven methodology**

Primarily used in the clinical trials and medical field, the Cox's regression, also known as the proportional hazards model, can be explicitly modelled by means of a probabilistic survival function. Cox regression analysis can be analysed through time to event occurring. For example, set = p(T > t), the probability that the patient survives more than t years. If mortality is the outcome variable, then one speaks of survival analysis. If F(t) =1- S(t), and J(t) = F'(t) is the first derivative of the distribution function F, then the concept of hazard, defined as h(t) = J(t) / S(t), gives the instantaneous risk of demise after time t. Logistic regression and Cox's regression are multivariate statistical regression methods (Uckun et al. 2008). Because this research is about monitoring failure occurrences, survival time using Cox's regression can be used to reflect failure events. Another example of data driven method is the state estimation techniques such as Kalman filters or other various tracking filters that perform the same function. Using this approach, the filter is considered to be a virtual sensor, whereby it provides optimal estimation of quantities of interest that may not be obvious. It uses knowledge of noise to minimise estimation error covariance by Kalman gain. Typically, Kalman filter is implemented using the linear system model but can be extended to non-linear model if desired.

2.8.3.1 Advantages and drawbacks

Although data driven method of prognostics is able to learn models based on empirical values, it requires an extensive fault history data. It is possible that this method provides the best solution if large enough data is available for analysis. This method uses historical data to automatically learn system behaviour and their degradation patterns. It suffers when insufficient or no data exist for analysis.

2.8.4 **Probability-based methodology**

Markov Model is used to illustrate a probability based method of prognostics that is used to allocate spares in the circumstance of any event when failures occur. The Figure 2-6 below shows the Markov model of the failure and repair process of a component in the presence and absence of spares. Here, it is assumed that the time-to-failure is exponentially distributed. When a component fails, it is immediately replaced with a spare if a spare is available, otherwise, additional spares will need to be procured. One disadvantage of using the Markov model is that as the states gets more complex, it get really tedious in solving the Markov model vector since the number of states in the Markov model usually grows exponentially with the number of system components. However, it is quite good for application where only the behavioural events are needed to be analysed and not the real physical system. Nevertheless, with the use of software simulation, the Markov model should not be that complicated to implement.

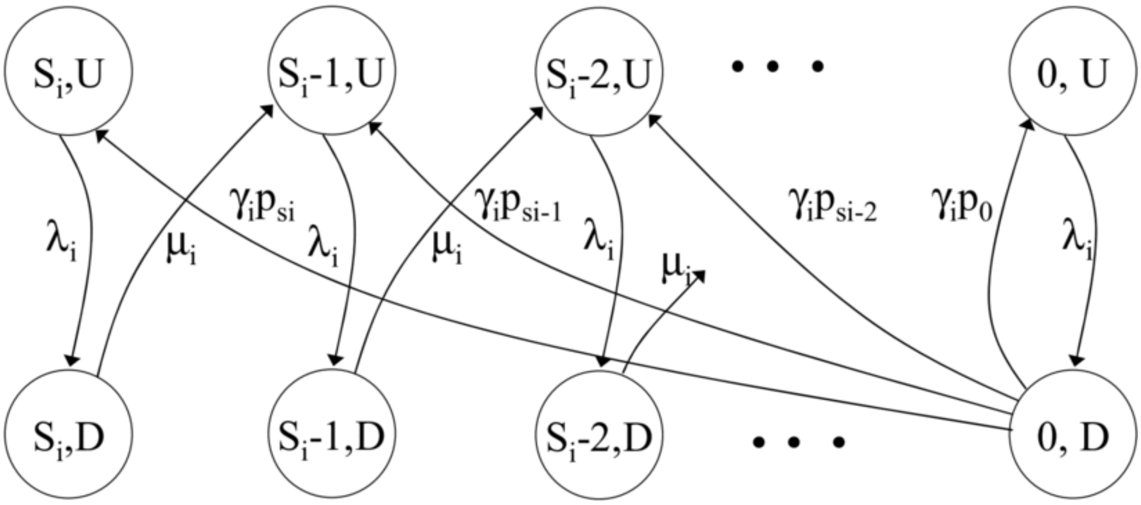

Figure 2-6: Component level availability model (Fang Tu et al., 2007)

2.8.4.1 Advantages and drawbacks

In probability-based methodology, which uses historical or sequential data to predict future failing, the main disadvantage is that it tends to have 'diffusion of context' phenomenon which brings context to generalisation. In contrast, the main advantage of probability-based analysis is that analysis can be made or tested based on probable outcomes. The common probability method will be elaborated in the next section.

2.8.5 **Bathtub curve and constant failure rate**

The bathtub curve is common when discussing reliability issues. A typical bathtub curve can be represented as shown in Figure 2-7.

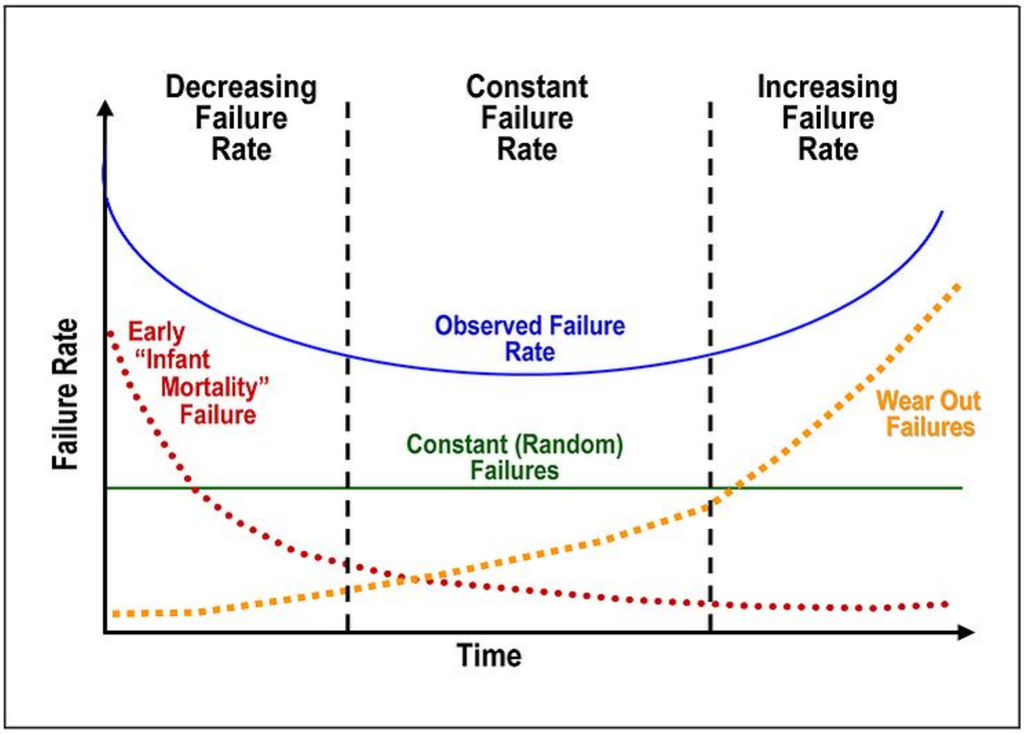

Figure 2-7: Typical bathtub curve used in reliability

Early method in reliability prediction uses constant failure rate (CFR) and reliability distribution using exponential. MIL-HDBK 217, which is the "Reliability Prediction of Electronic Equipment", is a military handbook. It has been so-called the industry standard for many years and uses CFR as its basis in reliability prediction. CFR has been used for many years and until now it has presented the basis of any true model as the physics of failure data is much harder to get hold of. Thus, it is still valid to consider using failure rate in the research as data that is obtained from the maintenance industry are still assuming constant failure rate to a certain extent (White, 2008). Airborne avionics systems apply the bathtub curve (LOU et al., 2009) which relates failure probability and time as shown in Figure 2-7. However, with electrical and electronic components that are ever-present to deal with, most semiconductors are said to have no short term wear out phase. This means, that the curve remain relatively flat as shown in the middle phase. It refers to useful life stage failure rate that is constant.

Constant failure rate usage in research is disputable. However, that has been the foundation stage for any process as true model based, physics of failure data is really hard to get. Failure data is commercially sensitive in the sense that if the rate is high, consumers will lose their trust in the product. That is why most manufacturers prefer to keep it confidential. Moreover, very few failure mechanisms have an established failure signature (Hecht, 2006).Thus, this 'forces' academicians to use whatever resources available such as online data and product specifications with failure rate numbers attached. This is because in normal consequences, reliability data will be published.

2.9 Overview of methodology

The methodologies used in this prognostics studies are threefold; the model-based method, the data-driven based method, and the hybrid method that combines both model-based and data-driven methods. Since a prognostic research involves estimation and prediction, statistical and probabilistic studies cannot be totally excluded. Thus, it is equally important to also consider statistical method and probabilistic elements in this research. Take a weather forecast as an example, to forecast tomorrow's weather, information must be known beforehand. In doing so, initial data has to be collected or certain pattern of weather forecast needs to be established. Otherwise, there must be a certain model that can be used to predict the weather. It could be by looking at the wind direction, or the moisture level and even the location of the clouds. Similarly, in forecasting the failure of aircraft equipment, several procedures are established. Firstly, historical data are collected. This will be the starting point that acts as the benchmark. Data is then analysed through models. Prognostic methodology model is developed to assist in estimating time-to-failure.

In this research, the mathematical model of this methodology employs the Markov Model and Cox's regression analysis incorporating well-known reliability standards which are common for space and military use. Various numerical methods for efficient and accurate solving of the model equations are presented, which enables reliable predictive simulation of the underlying physical phenomena. The simulation results are compared with the corresponding field data results and checked on their physical soundness. Details on the performance of the algorithm developed will be shown in later chapters.

2.10 Summary

This chapter has identified the research gap from the past researches that can be filled by this study. Through this chapter, trends of methods and application in the current practise can be seen. Besides the emerging methodologies, the advantages of applying prognostics methodology in the avionics context has been presented. An in-depth discussion on the methodology used for this research will be covered in the next chapter.

CHAPTER 3

RESEARCH METHODOLOGY

This chapter illustrates the steps that put this research in entirety. It brings together the process of achieving the objectives in fulfilling the aim of this study. This chapter will tell the reader the framework and structure of the avionic prognostics methodology. Each functional module reflecting the objectives of the avionics prognostics system will be thoroughly explained in this chapter. A pictorial representation of the whole process in given in the Figure 3-1:

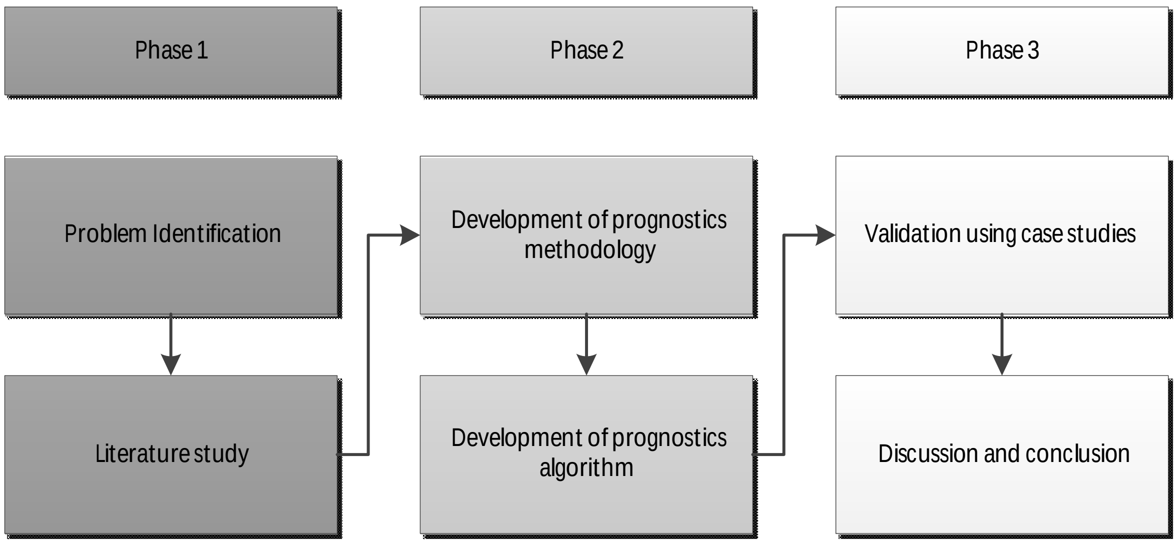

Figure 3-1: General steps of the research methodology

3.1 Problem identification

The early stage of this research was to identify issues relating to prognostics approach and how can it contribute to the current situation. The research has been narrowed down to prognostics application on avionics at system level due to the reason that most discussions have neglected this area of study. This particular research is aimed at exploring and developing prognostics methodology for airborne avionics system.

3.2 Literature study

Along with the aim, the literature study provides evidence on the importance of conducting the research. An extensive literature review has been carried out to enable a decision on the right methodology and approach to be chosen for LRU level prognostics application. Literature study was done continuously throughout this duration of study and is considered the most fundamental step in achieving the aim and objectives of this study.

3.2.1 **Steps in literature study**

a. Searching for literature

b. Sorting and prioritising the retrieved literature

c. Analytical reading of papers

d. Evaluative reading of papers

e. Comparison across studies

f. Organising the content

g. Writing the review

3.3 Development of prognostics methodology

In this research, the discussion on prognostics methodology is divided into two main sections:

- The general view
- The integrated methodology view

In general, the prognostics methodology will discuss the fundamental need for prognostics work. The process flow of this methodology is shown in the Figure 3-2:

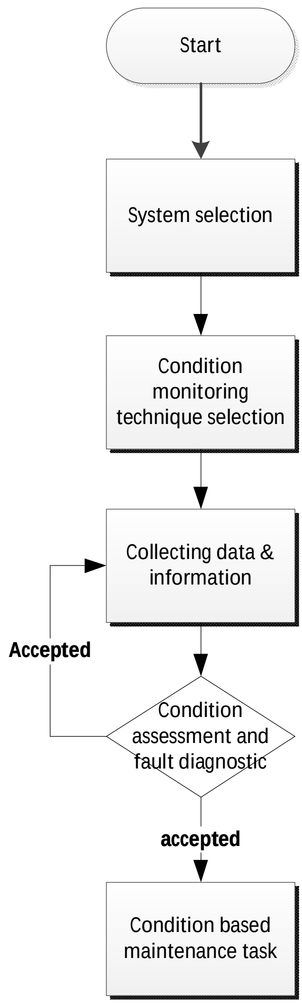

Figure 3-2: General process flow for prognostics based on condition-based maintenance

Identification and selection of condition parameters are necessary to ensure prognostics success. Only measureable parameters and parameters that could be monitored to define their condition or performance are to be chosen. These condition parameters can be defined as a measureable variable that enables to be displayed directly or reflect indirect information about the condition of an item at any particular instance. In practise, there are two distinguishable types of condition parameters which are the relevant condition indicator (RCI) and the relevant condition predictor (RCP). The RCI is a parameter that describes the condition of an item during its operating time and indicates the condition at the instant of inspection. The RCP on the other hand, describes the condition of an item at every instant of operating time. The difference between these two conditions is that the RCI is usually related to the performance at the point of inspection and not able to predict the future development of the considered system. RCP on the other hand, represents the condition of the system which is most likely to be affected by a gradual deterioration failure such as wear and crack growth (Kumar et. al, 2012).

The methodology includes several approaches such as Markov model theory, statistical analysis, mathematical model known as MTBF (Mean Time between Failures) and MTBUR (Mean Time between Unscheduled Repair) equations, and Cox's regression analysis that will be integrated in the final stage. Given the input such as in Figure 3-3, the specific results of prognostics methodology can be achieved. For example, if fault tree and failure rate are given or known, Markov Modelling can be used to determine the probability of failure at any level of analysis. As such, time to failure can be compared with the established failure rate by the OEM. Because prognostics study focuses on the importance of time in maintenance, all these approaches will involve the time factor. At the end of these processes, these outputs will then be synchronised using the temperature-failure rate model to calculate the probability of failure at different operating temperatures.

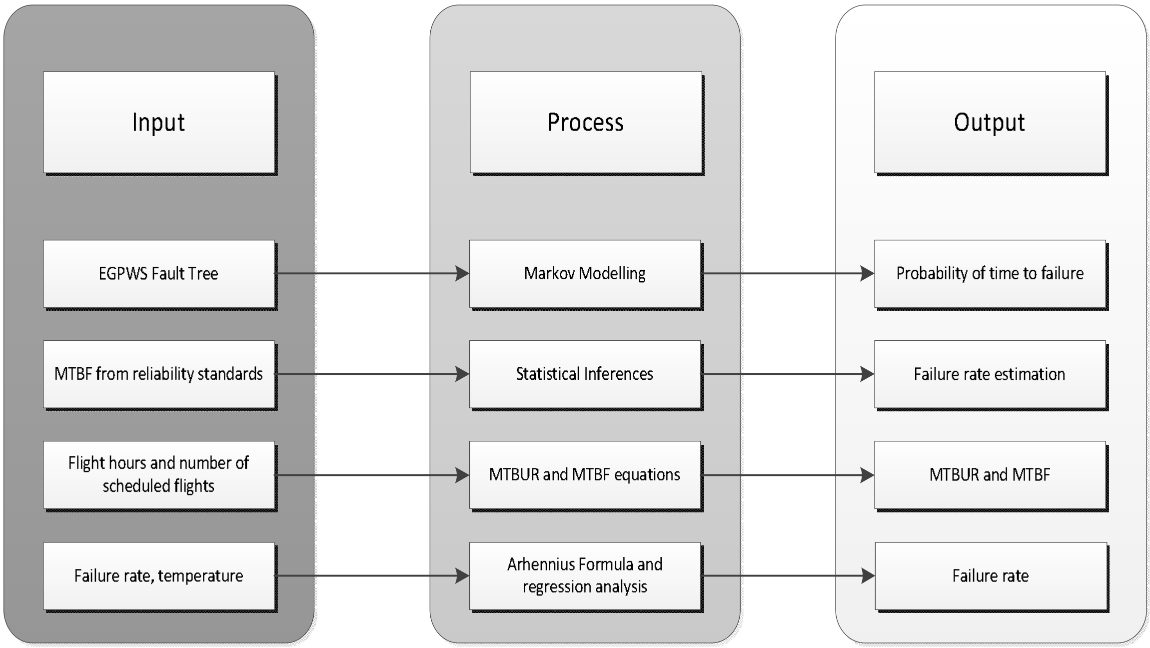

Figure 3-3: Prognostics methodology design process

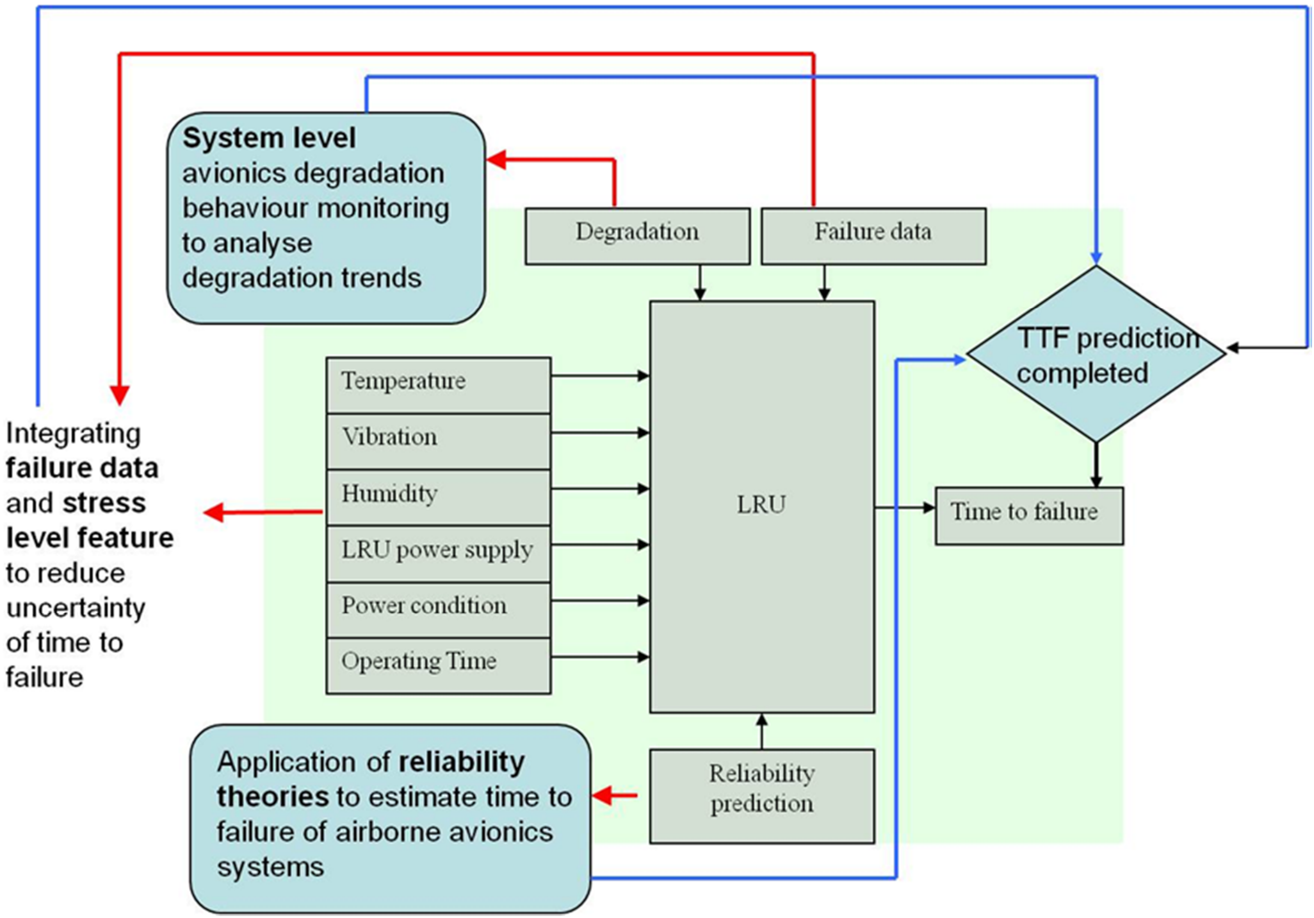

Figure 3-4: The integrated view of prognostics methodology

Figure 3-4 shows the overall prognostics methodology developed to find the time to failure in an LRU. It will be elaborated further in the sections that follow.

3.3.1 **Degradation**

In this part of the study, degradation trend analysis at system level was done by gathering data from airlines. Data includes component removal information with dates, types of aircraft, airlines, flight hours and ATA chapter description. In doing the analysis for dependency trends at LRU level in airborne avionic system, fault tree diagram with failure rate relationship was used.

3.3.2 **Failure rate data**

Failure rate data is used and produced at different stages of this research depending on the availability of information. In particular, failure rate data will be used in the three stages of prognostics methodology developed. First, it will be used in the Markov model process where failure rate is essential for simulating using the procedures developed. The failure rate dependencies within different equipment pooled for single failure rate for overall LRU will be calculated. Secondly, MTBF and MTBUR calculation are analysed, and failure rate is established. Thirdly, it will be used in the regression analysis of failure rate and temperature dependency analysis.

3.3.3 **Time to failure prediction**

Finally, all the information gathered and calculated will be integrated with the environmental (temperature) versus failure rate model to produce the time to failure prediction.

3.3.4 **Reliability prediction**

In this research, reliability prediction used is closely related to application of reliability theories such as probability of failure, MTBF, MTBUR, MTTF and failure rate. Reliability prediction was conducted to identify the relationship of the field data and the common known variables, and is a common methodology where prediction of failure is concerned. The intended results of this process are to fill in the gap of knowledge and to verify that the developed methodology works well.

3.4 Development of prognostics algorithm

An elaboration on the prognostics algorithm development is based on each method. The algorithm shall follow the Figure 3-5:

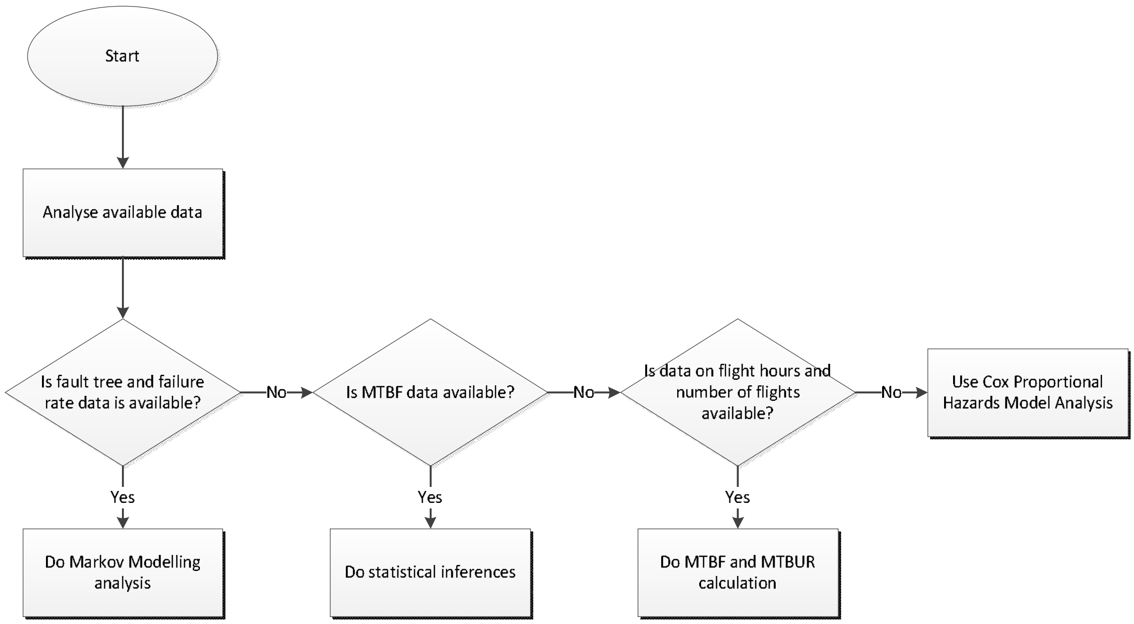

Figure 3-5: Overall prognostics algorithm development process

The algorithm starts off by analysing all available data and identifying what methods work best. If information on failure rate data is accompanied by the fault tree diagram, then Markov modelling can be used. Otherwise, the process flow continues to check if the MTBF data is available so statistical inferences can be done. If not, the system continues to check if flight hours and number of flight records are available. If they are, the MTBF and MTBUR data to compare against the OEM benchmark value can be calculated.

3.4.1 **Markov-Failure rate module**

This algorithm is used to solve probability of failure given the fault tree (degradation) and failure rate at each state. With the known data, the probability of failure at the top level can be calculated and compared with the OEM's value. Besides, dependencies of components in LRUs can also be figured out using the fault tree diagram. The detail of this process is given in the Figure 3-6:

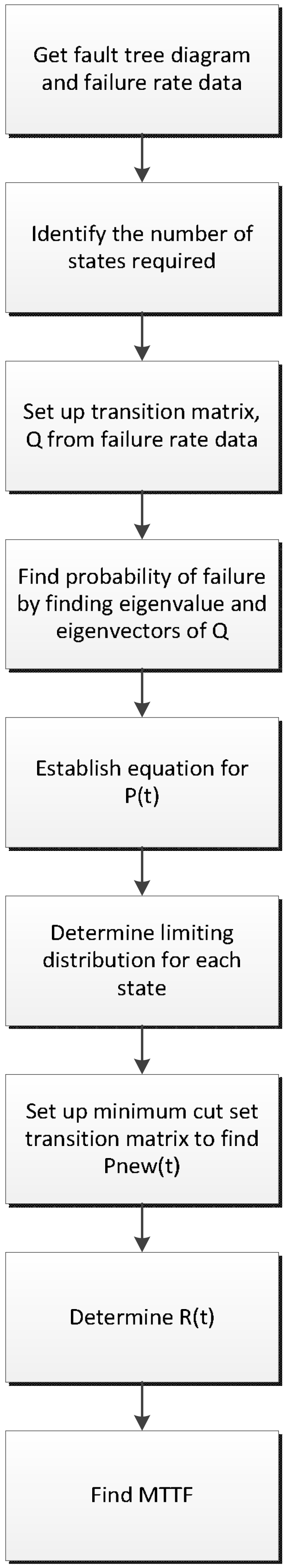

Figure 3-6: Algorithm for Markov Modelling

3.4.2 **Statistical inferences module**

Statistical inferences were used to explore and understand the pattern and trend of the removal of components. The result of the study provided insights to identify suitable components to explore and determine why the chosen component was selected.

3.4.3 **MTBF and MTBUR module**

The input parameters needed for this module represented in Figure 3-7 are:

- UR – unscheduled removal
- URY – unscheduled removal for the period of study
- IU – installed unit
- FH – flight hours
- CF – confirmed defect

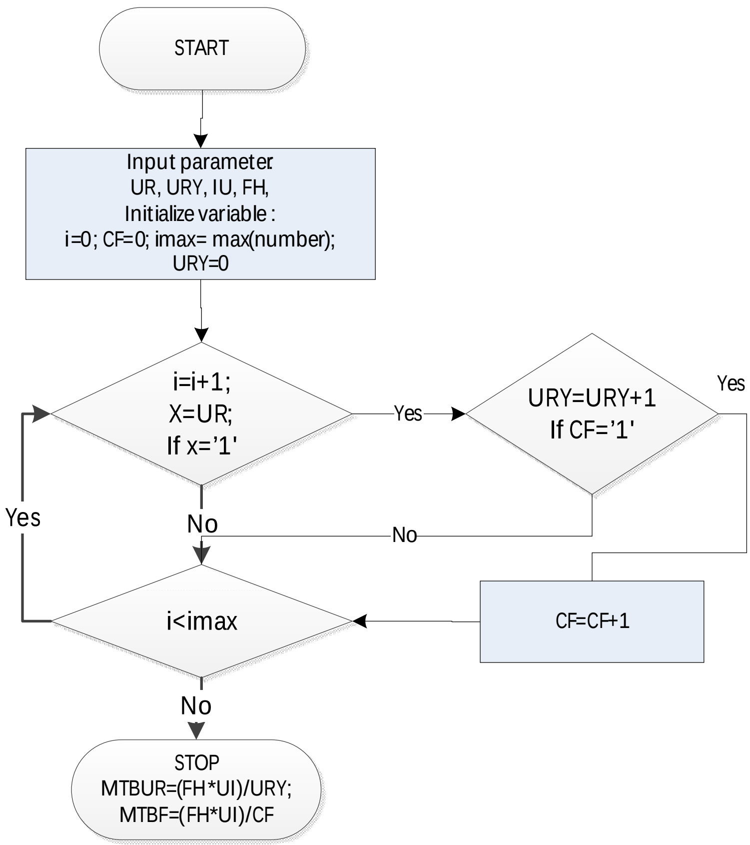

Figure 3-7: MTBF and MTBUR algorithm

This module is used to calculate MTBUR and MTBF values for removal data provided information of flight hours, confirmed defect, unscheduled removal data, and formulas are provided. For this module, the equations to be used to calculate MTBUR and MTBF values are:

- $MTBF = \dfrac{flight\_hours \times units\_installed\_per\_aircraft}{number\_of\_confirmed\_failure\_during\_that\_period}$
- $MTBUR = \dfrac{flight\_hours \times units\_installed\_per\_aircraft}{number\_of\_unscheduled\_removal\_for\_that\_period}$

3.4.4 **Cox Proportional Hazard analysis module**

For Cox proportional hazards model incorporate the effects of covariates which will be temperature and stress on failure rate values. Proportional hazards assumption will be:

$h(t;x,\beta) = h_0(g(x,\beta)$

where

$h(t)$ and $h_0(t)$ represents the failure rate;

x represents the covariates; and

β is the coefficient estimates.

Generally, four steps will be needed for this feature to be included in analysis.

- Step 1: Load sample data
- Step 2: Find the coefficient estimates
- Step 3: Add temperature and stress as covariates to the model
- Step 4: Analyse model for outcome

3.5 Validation using case studies of real field data

For validation, a case study is used. The selection of avionics system to be used as a case study is determined to fulfil the effectiveness and the need for prognostics application on such system. Thus, it must affect the safety, reliability and maintainability of the aircraft that the system employed. For that, the Terrain Avoidance Warning System was chosen as a case study. This decision was also based on the trend of removal, which has been identified to be amongst the most crucial, where improvement is needed.

3.5.1 **Approach to identify suitable avionics equipment**

In order to identify and evaluate the suitable avionics equipment to be applied with prognostic techniques, a large and diverse research of the component data was required. First, this thesis will focus on avionics systems installed in general transport aircraft. The application of avionics systems and components to be analysed are randomly selected from the maintenance manual. Next, the selected avionics systems are extracted into two main categories that are communication and navigation. The categorized avionics systems will be evaluated into several criteria to identify whether or not the systems or components are worthy of prognostics studies or vice versa. The factors for considering if systems are prognostics worthy are cost, operation, logistics or replacement issues and lastly the maintainability of the system or equipment as summarised in Figure 3-8.

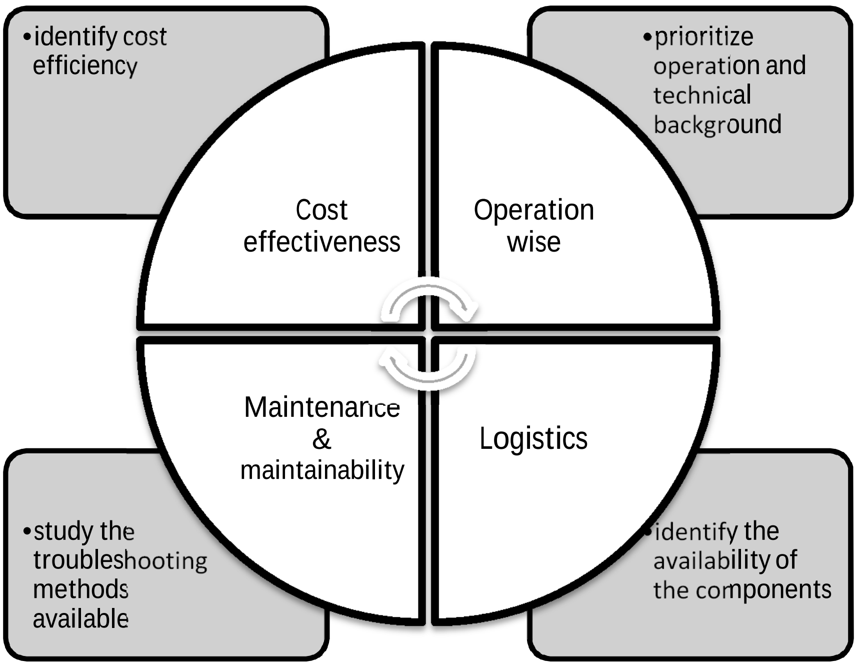

Figure 3-8: Avionics metrics parameter

3.5.1.1 Cost-effectiveness

Cost effectiveness is an economic metric that measures the average cost to repair a component during a year. Theoretically, cost effectiveness decreases initially as repair sources advance up the learning curve and develop more efficient processes based on previous repairs. But as components age, repair costs will increase in real dollars. Factors affecting these increases include lower repair volume, diminishing repair sources, more expensive replacement parts, and more expensive test equipment maintenance costs. Measurement of this metric is the easiest with outsourced vendors who charge a service fee for each repair, whereas organic repair costs can be difficult to track if cost pools are not sufficiently separated.

This metric can be used to estimate the future component life-cycle costs for use in return on investment calculations in order to determine when component repair will become cost prohibitive. The time that component repair becomes cost prohibitive represents the end of a component's life cycle due to economic considerations. In addition, this metric is also useful for determining the short-term estimate of repair budget.

3.5.1.2 Operation

This metric describes the operation of the avionics system and how critical it is in ensure the reliability and availability of the aircraft is achieved. This metrics portray the knowledge of the system and demonstrates how the equipment operates in real life. The priority of the avionics system to be prognosed is greatly dependent on the criticality of the equipment in order for the aircraft to be airworthy. Besides that, as the knowledge operation of the equipment is achieved, it is easy to differentiate the criticality each of the systems towards flight safety.

3.5.1.3 Logistics for availability

This is a sustainability metric that measures the percentage of components in stock that are required to be in stock due to an allowance list. This metric is also referred to as stock age rate or supply rate, and can have several methods of measurement, any of which can be used in the model.

System availability actually measures a state, such as an end of the month snapshot, as opposed to a rate or flow. For example, if there were ten operating sites that each has an allowance of four components, a supply of 40 units at the end of the month would equal 100% availability. This measurement would be difficult to measure continuously in real time over the course of the month. However, the average of the 12 monthly measurements would be a suitable yearly entry into the model.

This metric measures the ability of the repair system to provide an adequate supply of operable components, but there are some inherent shortcomings to this measure. For one, since a goal of 100% is neither obtainable nor desired, it is difficult to determine a suitable goal. Also, if 50% is measured during one month, it could be due to ten users having two out of four components available, or it could be due to five users having no components available at all. Therefore, this metric does not track the distribution of the availability very well.

Recent efforts have sought to reduce the capital tied up in large stockpiles of repairable parts, at operating units, by reducing repair pipeline cycle times and transportation delivery times. Both of these efforts have decreased required on-site supply needs. But if a component is experiencing sustainment difficulties, such as diminishing repair sources and stocks of piece-parts, obsolete test equipment, and high scrap rates, availability will decline. The increase in repair pipeline and supply cycle time will prompt an increase in allowance limits that will reduce availability even further. This can however be improved if the procedure of AOG was properly used and could supersede issues discussed.

3.5.1.4 Maintenance and maintainability

Maintenance optimization is a process that attempts to find the best balance of the maintenance requirements (contractual, economics, technical, etc.) and the resources used to carry out the maintenance program (people, spares, consumables, equipment, facilities, etc.). When the maintenance optimization is effectively implemented, it will improve system availability, reduce overall maintenance cost, improve equipment reliability, and improve system safety. In the former case, optimization is performed to choose the option that generates the largest cost avoidance and maximizes the availability for an individual system. In the latter, optimization is performed to choose the optimal subsystems to be maintained and meet availability at the enterprise level.

The maintenance efficiency of systems is an important economic and commercial issue. Main difficulties are resulted from the choice of maintenance actions. A bad choice can lead to maintenance with a cost overrun that is not acceptable. Because of the increase of involved technologies (pieces of hardware, software) and the different interactions between components (communications by message passing or physical interactions), the decision of a maintenance action is very complex and requires a diagnostic and prognostic analysis. Maintaining such a system basically consists in replacing components that are unable to perform their function by new ones.

Maintenance activities are costly for several reasons. The first one is that they usually require stopping the system that cannot be used anymore during the maintenance phase. The longer the maintenance phase is, the more costly it is. It follows that the maintenance phase must be reduced to the strict minimal operation that is the replacement of the correct components. This requires that the maintenance actions must be decided relying on an efficient and complete analysis of the health of the system when it is operating. The second reason of a high cost in maintenance is in cases of emergency. If a component suddenly fails and the system fully breaks down, it automatically requires some unscheduled maintenance actions, which are more costly than scheduled maintenance. To partly avoid this issue, prognostic methods are used in order to perform preventive maintenance. Preventive maintenance basically involves replacing components during a scheduled maintenance phase that are not yet faulty but that will inevitably become faulty before the date of the next scheduled maintenance phase.

3.6 Summary

This chapter details out the methodology chosen for this study and provides understanding to the reader on why the steps of methodology were identified and used in the study. In the next chapter, theoretical work on the study will be presented.

CHAPTER 4

RUL PREDICTION METHODS IN LRUs

This chapter aims to provide a systematically developed fundamental theory of the research models based on the study done in the earlier chapters. The Remaining Useful Life Prediction methods involved in carrying out this research will also be provided in this chapter. It will include both the RUL prediction methods for LRUs and LRMs as well as the RUL prediction methods for airborne avionics system. The proposed prognostics algorithm of the prognostics system will be further evaluated through the development of software simulation system.

4.1 Fault tree analysis

Fault tree analysis is a failure based study that includes logic and probabilistic techniques. This application, which was popular in the sixties, is particularly preferred in manuals and instruction booklet as it is straightforward and clear. It is a qualitative approach but can be quantified when probabilistic risk assessment is added. Thus it becomes a mixed approach. It has become a useful methodology in system safety assessment, where all failure rates are presented in logic diagrams. Top events can be easily seen when illustrated in a fault tree diagram. With fault tree, a deductive approach is used to conclude what events trigger failure to happen. Basically it is a top down approach. The process determines the root causes using organised backward steps design, to find the underlying solution of the overall failure. The advantage of using this method is that it will not only show the low-probability and high consequence failure events but it can also show high-probability and near miss events. It is the benign events that are important to detect as high-consequence failures can blow up when are not detected early. Fault tree uses logic diagrams such as the AND gate and OR gate as in Figure 4-1 in order to describe the input and output events. This relation of fault tree is commonly associated with flight hour and thus possible to find the mean time to failure when the failure rate is known.

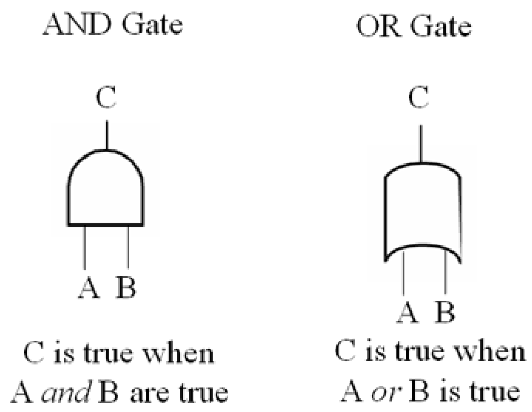

Figure 4-1: AND and OR gate used Fault Tree Diagrams

4.2 Markov model

Markov model is a stochastic process that involves a probabilistic mathematical model, which involves time, and its outcome only depends on the present state. This means, the next state outcome is only influenced by the preceding state. Since Markov model can be used as discrete or continuous processes with regards to time, it has been used in many areas of reliability. A Markov model consists of two variables, which are state and time. Normally, Markov is represented in the form of state transition diagram as shown in the Figure 4-2.

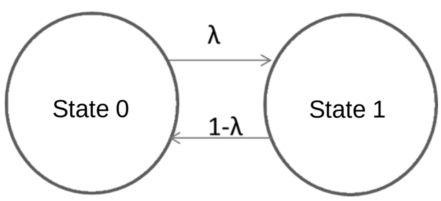

Figure 4-2: State transition diagram

As shown, a component with two states, normal state and fail state would have this as its state diagram or transition diagram. The transition rate is described by the failure rates in this study. The one-step transition probabilities can be condensed into a transition probability matrix **P**, where

$$\mathbf{P}_{ij} = \begin{bmatrix} P_{00} & P_{10} \\ P_{01} & P_{11} \end{bmatrix} \qquad \textbf{(Equation 4-1)}$$

The following are some important properties about Markov model:

- Since for all *i j* ∈ S $0 \le x \le 1$ and each row in **P** adds up to 1, matrix **P** is a stochastic matrix.
- The probability mass function of the random value ***P***(0) is called the initial probability row-vector
- $X(0) = [X_0(0), X_1(0), \ldots X_n(0)]$ and presents the initial condition of Markov Chain.
- If **P** is the state transition matrix, and X is the state probability in exponential Markov chain then $X'(t)=X(t).\mathbf{P}$

For Markov distribution model, for a given the initial distribution X(0), the following can be determined.

- $X(1)=X(0).\mathbf{P}$
- $X(2)=X(1).\mathbf{P} = X(0).\mathbf{P}.\mathbf{P} = X(0).\mathbf{P}^2$

Thus, for any k,

- $X(k)= X(0).\ \mathbf{P}^k$ and elements of **P** must satisfy the following conditions:
- $\sum_{j-1}^{j-n} P_{ij} = 1$ for all I (row sum) and $X_{ij} \geq 0$ for all i and j.

The dynamic nature of system is modelled as the Markov state model. The Markov model provided prognostic measures, such as the time to reach a faulty state, along with the probability of reaching this state. The Markov model allows the system to go back to their previous state and there is then no need to consider unidirectional system progress because electronic products do not experience failure due to wear out mechanisms.

In Figure 4-3 below, is the GUI for a simple two state markov chain simulation which shows the probability of system going into states S1 and S2. X represents the initial state distribution and P is the state transition matrix. Figure 4-4 shows a simulation for a 3-state markov chain where X3 steady shows the output when the system stables off (Source code provided in Appendix 2).

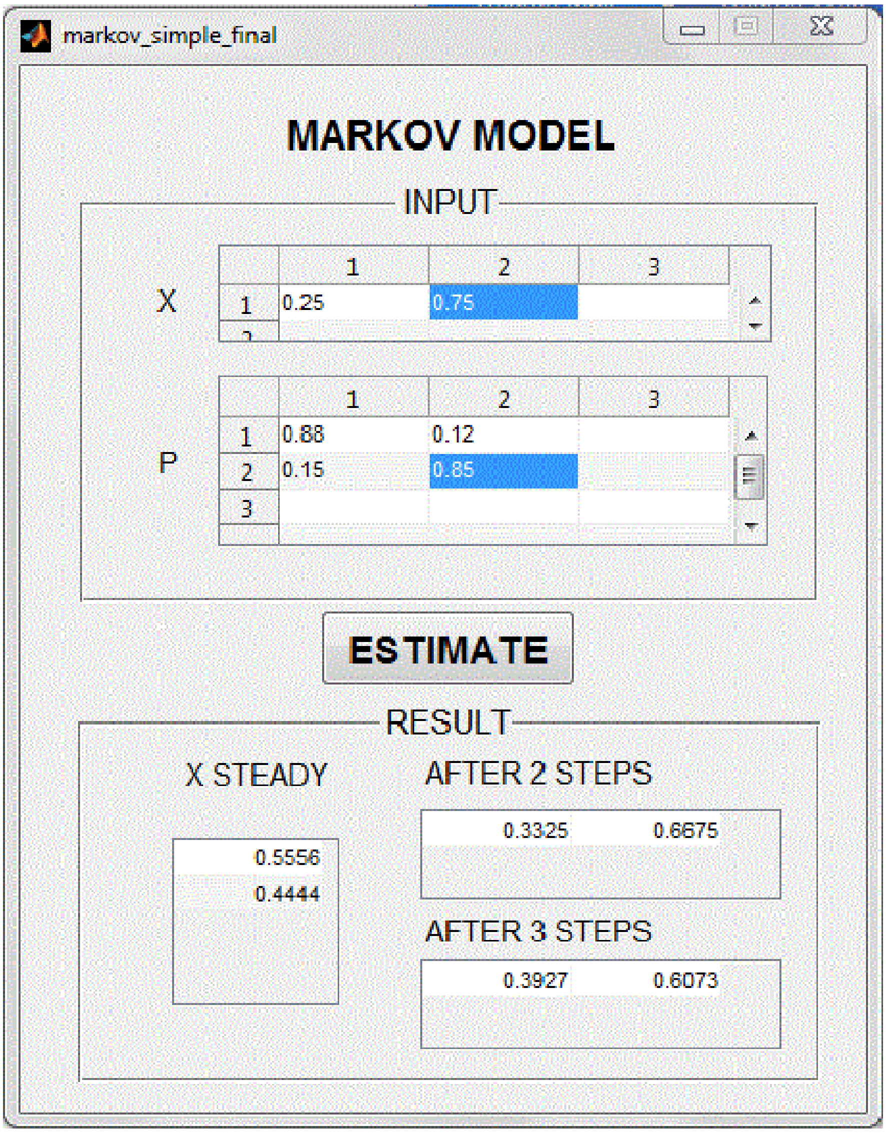

Figure 4-3: Two-state Markov Chain Simulation

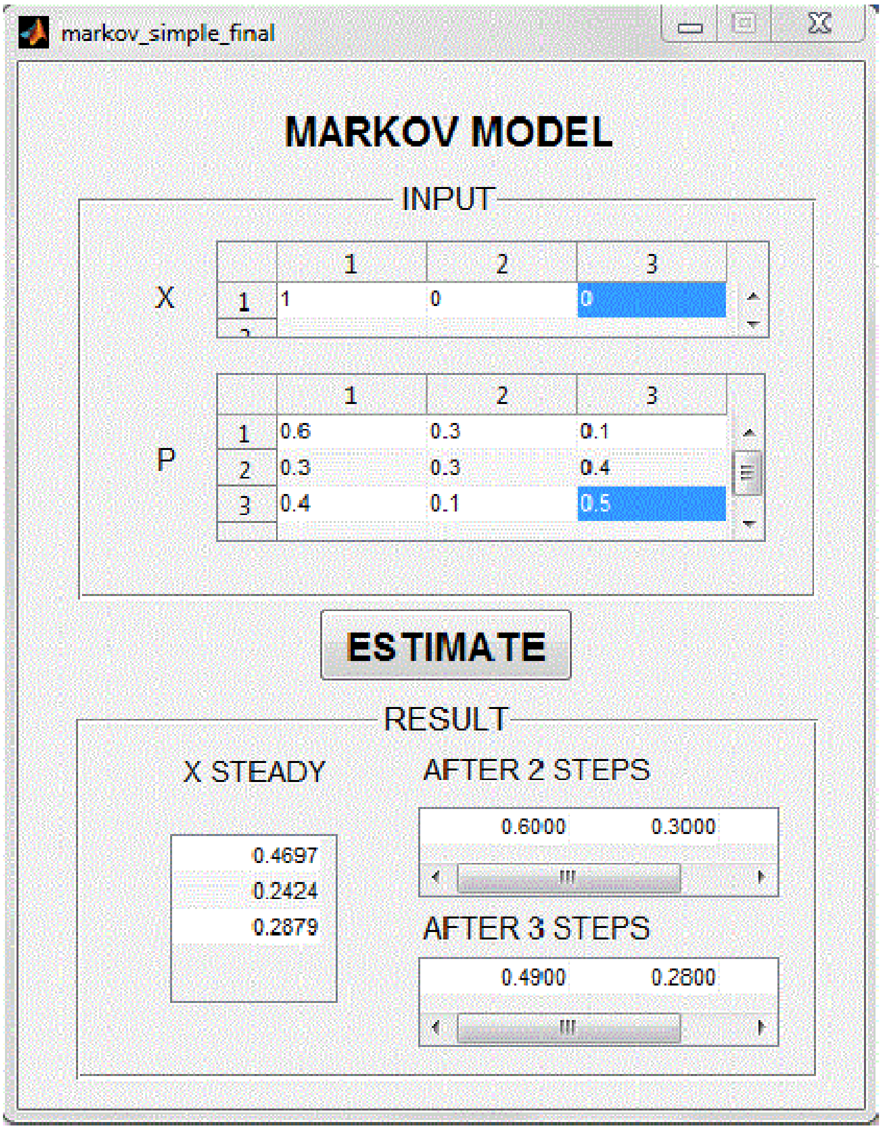

Figure 4-4: Three-state Markov Chaim Simulation

4.3 Cox regression analysis

Statistical procedure of Proportional Hazard model is used in identifying objective assessments in determining the true health state of a system. Cox proportional hazard model on the other hand is a multivariate technique for analysing the effect of two or more metric and or non-metric variables on survival. Failure condition at repair time is noted and failure time when the system cannot perform its function is recorded. Data that will be needed to perform this analysis will be system/ LRU fail time, removal due to failure or preventive maintenance due to signal of deterioration. Determining the probability of a system will fail given its initial characteristics and evolution over time relative to other systems. It also explores the timing of LRU going to failure. Therefore, a robust control using actionable information can be used for failure prognosis. It is useful when dealing with many covariates, X. The main advantage of using proportional hazards model is that analysis can be carried out without any assumptions about the distribution and about the form of the base line hazard function, h_0 (Dale, 1985). Given sample data, that contains two or more variables, model of the relationship can be derived between the variables. Next, it is then decided which model best describes the relationship between the variable and estimates its accuracy. In order to avoid overly optimistic prediction error is by doing cross-validation. That means two sets of data are needed, one to build the model and the other to test the model.

Proportional hazard function is widely used in medical field to perform survival time prognostic. It is basically a model free and is a semi parametric model that needs no assumption to be made about shapes of time to event distribution. It can be used for events that deal with failure time data. Estimation techniques of hazard function will be used to predict failure times. Usually failure times are modelled by fitting an exponential, Weibull, or lognormal distribution to the data. As failure data arise with certain degradation parameter, data of occurrences can be used to correlate these two or more variables by gauging the weight attributed to each variable respectively. Because this technique allows for both metric and non-metric analysis, this method would then be generic enough to be applied to a “black-box” system of any kind.

4.3.1 **Advantages of hazard model**

‘Acceleration model’ rather than a specific life distribution model lies in its ability to model and test inferences about survival without any prejudgement or specific assumption about the form of life distribution model. The real strength of this proportional hazard model is that it allows for survival time relationship to be modelled through hazard function. Cox’s regression model is a non-parametric approach to survival data. Users can also incorporate time-varying covariates or explanatory variable that change with time. For example, if the system degrades before failing, the hazard model will change and it will be revealed in the health of system checks. Explanatory variable or predictors, X can be voltage, degradation parameter or others.

This model interprets the benefits of parametric and semi parametric approaches to statistical inferences. Also known as the Cox model, it presumes that the ratio of the hazard rate to a baseline hazard rate is an exponential function of the parameter vector.

$$\lambda(t, z) = \lambda_0\ (t) exp(\sum(j = 1)\hat{}q\beta j\ \ X(j\ )) \qquad \textbf{(Equation 4-2)}$$

Where z is a vector and β

$$\frac{h(t)}{h_0(t)} = \exp(x'b) \qquad \textbf{(Equation 4-3)}$$

$$\frac{h(t)}{h_0(t)} = \exp(X'B) = e^{b_1 x_1 + b_2 x_2 + \ldots + b_p x_p} \qquad \textbf{(Equation 4-4)}$$

$$hazard\, ratio(t, x_1, x_0) = \frac{h(t, x_1, \beta)}{h(t, x_0, \beta)}$$

$$hazard\, ratio\ (t, x_1, x_0) = e^{\beta(x_1 - x_0)}$$

The failure data (part total hour) will be correlated using the equation above with λ_0(t) unknown and β unknown. X on the other hand, will be the parameters to be studied such as temperature, humidity and vibration. When the hazard is logged, the coefficients are called the risk score, represented by β. When β is positive, it means that the two variables are positively correlated with higher better representing higher correlation. Otherwise, if β is negative, it means the opposite. One other method to find β is solving the Partial Likelihood Estimation (PLE). PLE can be calculated using the steps below:

1. Order failure times such that *t1<t2<…<*t_k where t_i denotes failure time for *i*th individual

2. For censored cases, define it as ‘1’
3. Ordered events are then modeled as a function of covariates, *x*
4. Take the product of conditional probability of failure at time t_t, provided the number of cases
5. The results will then show the probability of the *j*th case will fail at time T

Last but importantly, traditional statistical approaches such as regression models for survival analysis will be used to correlate environment parameters without knowing its distribution. Cox’s proportional hazard model in particular was chosen to be used since no assumptions need to be made about the shape of time to event distribution (Shyur, 2008). It is also suitable for semi-parametric or non-parametric statistical models, which will be used for this study. Cox’s regression analysis were mostly used in the medical field, but recently, it has also gained popularity in areas such as reliability engineering, finance for bankruptcy estimation, transportation and also system failures in general. With this model, many parameters can be taken into account, which is considered an important section of this research.

4.3.2 **Advantages and disadvantages of Cox model:**

- Provide estimate (statistical technique) of behaviour or condition effect on failure time given their prognostic variables.

- Data needs to be fitted using a mathematical model and final model will output a formula for hazard as a function of several explanatory variables.

- To analyse the model, coefficients are examined. Positive coefficient for the variable dictates that hazard is higher which then means prognostic work is worsening. However, if negative coefficient is shown, it means that prognosis is better for the system.

- The disadvantage of this model is that it only simultaneously explores using available data and not directly model based using sensor of any type.

4.4 Kaplan Meier

Coit presented a similar way to do reliability prediction. However, the demonstration was done with 39 circuit cards with different operating conditions (Coit et al., 2005). The end product shows reliability versus time graph as shown in Figure 4-3.

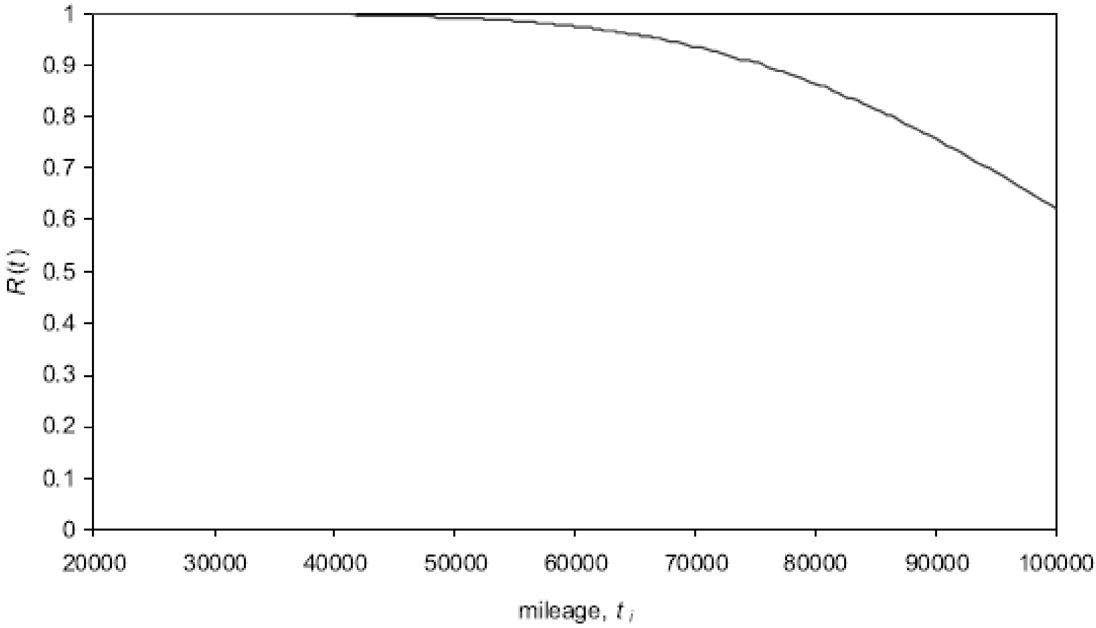

Figure 4-5: Reliability chart to determine time to failure (Coit et al., 2005)

However, they have assumed that the distribution of sample to be normal. Kaplan Meier is a method to estimate the cumulative survival distribution without making any distribution assumptions. It has proved to be an excellent use for large datasets and provides means to capture the lifetime distribution for 'snapshot' data. An output from the Kaplan Meier method can be shown in Figure 4-4. The disadvantage of using this method is it only provides an estimate of proportion of population that will survive and not truly accurate. However, for a simplistic view, this method seems to be practical.

Appropriate probabilistic model for time to failure is needed to be constructed and parameters need to be estimated so that the information can be suited to predict remaining useful life. Other usage is to establish inventory rules and also part replacement programmes. This technique will also help in the reliability program for the company. In order for prognostic technology to be applied successfully, the economic aspect of it is a priority. Broad prognosis indicator being the key identifier to any signature failure can contribute to cost saving (Hecht, 2006) . Many different prognostic techniques have been applied such as statistical methods, artificial intelligence methods and fuzzy-rule systems (Jianhui Luo et al., 2003). Prognostics, while has been established in automotive and power-plants, it is still quite new in application where many high failure rate parts are dominant with few recognisable failure mode.

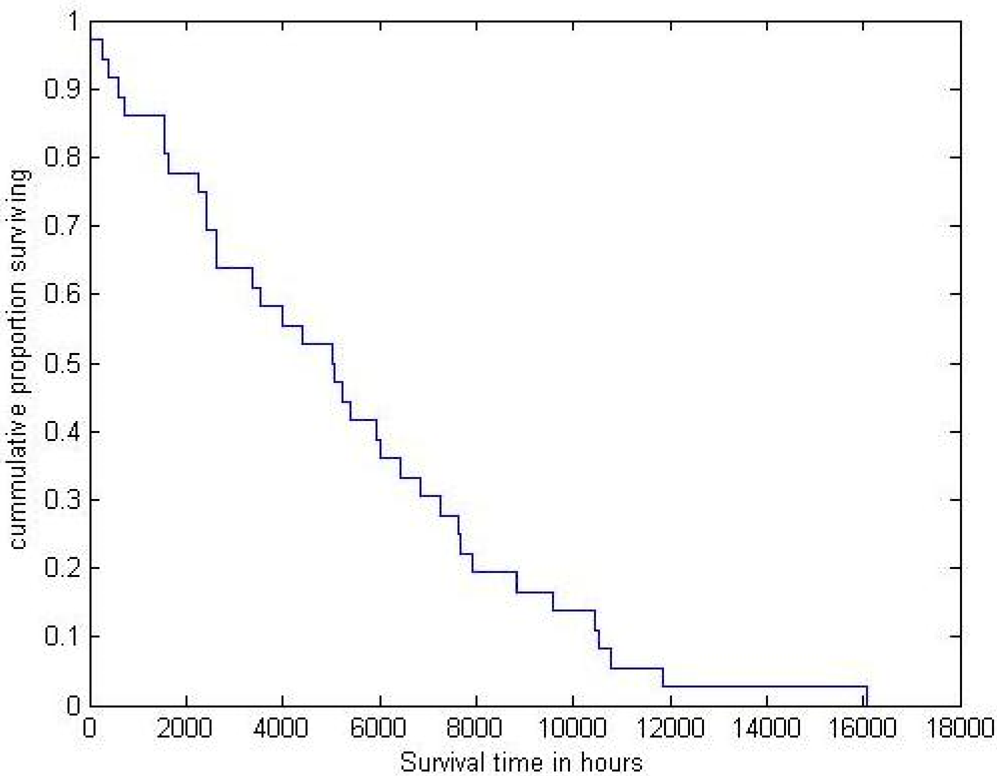

Figure 4-6: An example of graph produced using Kaplan Meier method

CHAPTER 5

TERRAIN AWARENESS AND WARNING SYSTEM (TAWS)

This chapter provides some understandings regarding Terrain Awareness and Warning System (TAWS) or sometimes referred to as Enhanced Ground Proximity System (EGPWS). The insights of this chapter cover TAWS/EGPWS functions and operating modes, TAWS/EGPWS system architecture and components, TAWS/EGPWS performance requirements and performance degradation pattern. Other than that, main fault mode and failure effects and how failure is propagated in the system will be analysed in this chapter. The reliability rates and failure rates will also be discussed as it will also be used for analysis in the later chapters.

5.1 Introduction

"ICAO's first action in this regard can be traced back to 1978, when requirements for equipping commercial air transport aircraft with GPWS were introduced in Part I of Annex 6 to the Chicago Convention. This led to a significant decrease in the number of CFIT occurrences, but not to their complete elimination. A further step was taken with the development of GPWS with a forward looking terrain avoidance function, generally referred to as enhanced GPWS and known in the United States as Terrain Awareness and Warning System (TAWS)."

-ICAO MODEL REGULATION AND GUIDANCE MATERIAL ON GROUND PROXIMITY WARNING SYSTEM (GPWS)

Enhanced ground proximity warning system was pioneered by Allied Signal now known as Honeywell. The main purpose of EGPWS is to provide basic ground proximity warning. Aircraft input such as position, altitude, air speed, glideslope and flight plan along with internal terrain and airport database allow EGPWS to predict potential conflict between the aircraft's future flight path and terrain. It is also to alert pilots on altitude awareness, excessive bank angle alert, terrain clearance, and terrain and obstacle awareness alerts. It is to warn the pilot of any inadvertent distance to the ground. It was before the use of enhanced ground proximity warning system (EGPWS) that occurrence of controlled-flight-into-terrain (CFIT) accidents was high. It was intended to reduce the incidents happening. These incidents happen with no signs of mechanical failure or fault but crashes to ground. These accidents usually occur in conditions of poor visibility due to atmospheric obscuration such as fog or rain, or darkness of night. Federal aviation regulations (FAR) have required installation of the system on large turbine-powered aircraft in commercial service since 1975. These system consist of a computer which gets inputs from sensors on aircraft and provides warnings to pilot. This is done through visual and aural alerting devices.

The system is designed to detect and warn the pilot of excessive descent rate near the ground, excessive terrain closure rate, approaching the ground with landing gear or flaps not in the landing configuration, and descending significantly below the ILS electronic glideslope when on approach to landing. Also, during take-off and immediately after initiating a missed-approach go-around, the system warns the pilot when the aircraft is descending when it should normally be climbing.

5.2 Hardware composition of the EGPWS (TAWS) and performance requirement of components

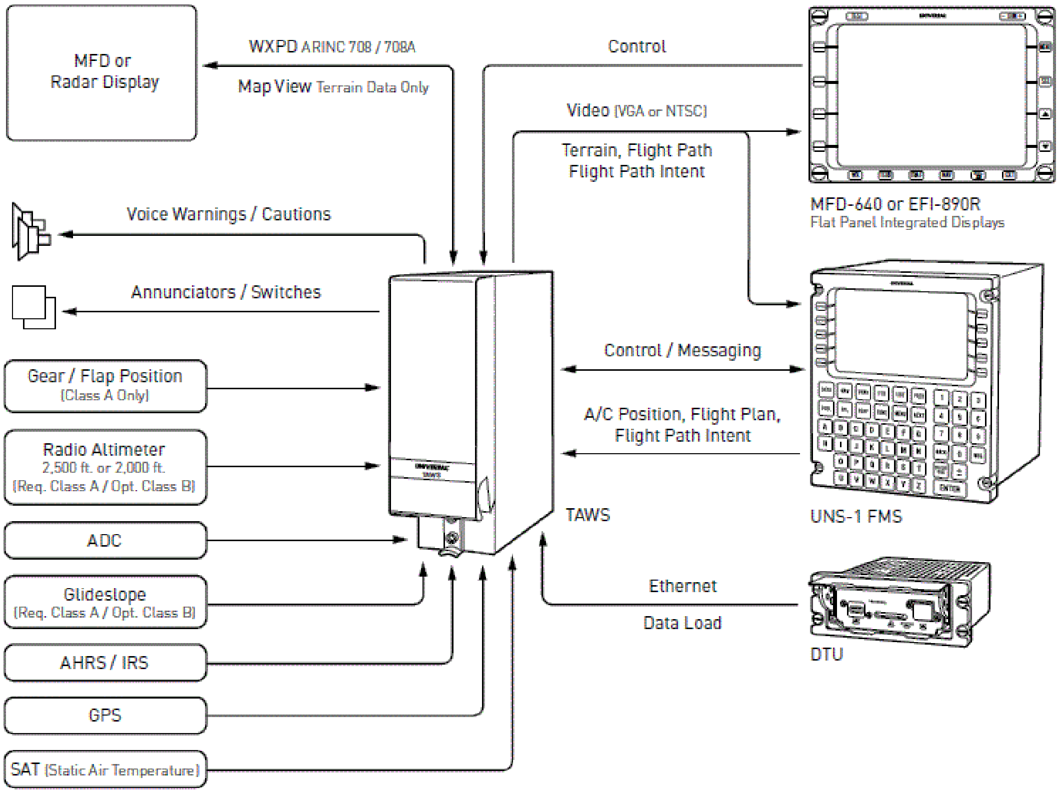

Figure 5-1: EGPWS composition

Figure 5-1 shows the composition on input going in and output leaving EGPWS. In the figure, outputs can be shown on the display and also through the voice warning for aural sounds. Some of inputs into the system includes reading from the radio altimeter, and the GPS integrated with data downloaded through the Ethernet and the FMS (flight management system.

5.3 Functions and operating modes of EGPWS

It is a system that warns the crew if the aircraft's current flight path would result in impact with the ground. The system is designed to capture the aircraft's flight path with respect to the terrain at all altitudes between 50 and 2450 ft. It uses inputs from systems providing radio altitude, air speed (Mach number), landing gear and flap position, and decision height (DH) setting. The system provides both visual alert message and aural alert warnings. The various dangerous conditions that can be encountered in flight are divided into six modes. They are mode 1: Excessive descent rate, mode 2: Excessive terrain closure rate, mode 3: Loss of altitude after take-off (or go-around) when not in the landing configuration, mode 4: Insufficient terrain clearance, mode 5: Descent below ILS glide slope and mode 6: Descent below selected minimum decision height (DH).

Advanced versions of the equipment have additional facilities of radio altitude callouts and aural warnings at excessive high bank angles. The other feature in these versions is that spurious and nuisance warnings are minimized. The system has a major drawback in that it cannot look ahead at terrain but can be integrated with a Worldwide Terrain database to give some look ahead prediction. However, this would introduce another failure mode which could be difficult to test. Consequently, it cannot always give pilots sufficient time to predict and plan avoidance manoeuvres. Enhanced GPWS (EGPWS), besides providing traditional GPWS alerting functions, displays the surrounding terrain (up to 320 NM) on an EFIS (electronic flight instrument system) screen or weather radar CRT (cathode-ray tube) and provides alerts about a minute's flight time or more away from terrain.

5.4 EGPWS system architecture

The Figure 5-2 shows the functional diagram of an EGPWC, with its inputs and outputs from the EGPWC.

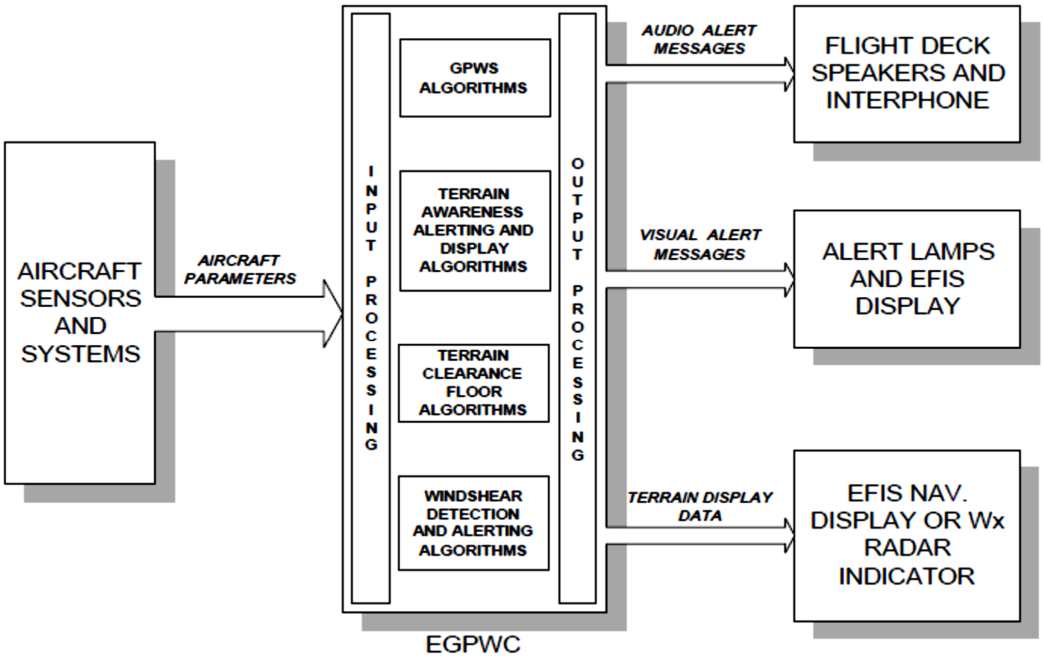

Figure 5-2: Overview of EGPWC and its components

5.4.1 **Aircraft sensors and other system**

The sensors and other system provide input signals to be read into the EGPWC for processing.

5.4.2 **EGPWC**

The EGPWC is the heart of the system. It computes and analyses the data provided to be fed out to the speakers and interphone.

5.4.3 **Flight deck audio systems (speakers and interphone)**

This is one of the outputs which provide sound alert to warn when necessary.

5.4.4 **Alert lamps and/or digital outputs to EFIS displays**

Lamps and EFIS displays are for alert and system status messages.

5.4.5 **Weather radar indicator or EFIS displays**

Weather radar indicator and EFIS displays provide display of terrain.

5.4.6 **Switching relay(s) or display switching unit when required**

This unit shows switching display inputs from weather display to terrain display.

5.5 EGPWS performance degradation

The EGPWS provides a Self-Test capability for verifying and indicating intended functions. This Self-Test capability consists of six levels to aid in testing and troubleshooting the EGPWS. These six levels are:

Level 1 – **Go / No Go Test** provides an overview of the current operational functions and an indication of their status.

Level 2 – **Current Faults** provides a list of the internal and external faults currently detected by the EGPWC.

Level 3 – **EGPWS Configuration** indicates the current configuration by listing the EGPWS hardware, software, databases, and program pin inputs detected by the EGPWC.

Level 4 - **Fault History** provides an historical record of the internal and external faults detected by the EGPWC.

Level 5 - **Warning History** provides an historical record of the alerts given by the EGPWS.

Level 6 - **Discrete Test** provides audible indication of any change to a discrete input state.

A level 1 Go/No Go Test is normally performed by flight crews as part of pre-flight checks. All other levels are typically used for installation checkout and maintenance operations.

5.6 Reliability of EGPWS from product specification

The EGPWC Failure Modes, Effect and Criticality Analysis (FMECA) was developed using MIL-STD-1629 as a guideline. The EGPWC reliability prediction was developed using MIL-HDBK-217F as a guideline. Historical MK

V GPWC reliability data and the EGPWC reliability prediction results were used as baseline criteria in establishing the following minimum EGPWC Mean Time between Failure (MTBF) and Mean Time between Unit Replacement (MTBUR) values. MTBF for confirmed failures will be 10,000 flight hours or better, for the latest EGPWS configuration three years from initial production delivery. MTBUR will be 7,000 flight hours or better, for the latest EGPWS configuration three years from initial production delivery. The MKVII EGPWS MTBUR is expected to be similar to that of the MKV EGPWS provided proper line troubleshooting procedures are followed when diagnosing system failures.

Historical MK V GPWC as well as recent Enhanced MK V field reliability data and MIL-HDBK-217F were used as baseline in establishing the minimum EGPWC MTBF and MTBUR values. MTBF values are per operating hours for confirmed failures and apply to corresponding latest EGPWC configuration three years from Initial Production Delivery (IPD) date. Similarly, MTBUR values are per operating hours for justified removals and apply to corresponding latest EGPWC configuration three years from IPD date. MTBUR values presume proper line troubleshooting procedures are followed when diagnosing system failures.

5.7 Failure rates of EGPWS components

5.7.1 **Failure rates standard from product specification**

Table 6-1 describes the EGPWS failure rate obtained from product specification that is used as a standard or benchmark for comparison with EGPWS failure rate obtained from field data.

Table 5-1: EGPWS LRU failure rate from product specification

LRU	Failure rate	MTBF (hours)
Radio Altimeter	$189.5x10^{-6}$	5277
Vertical Gyro	$247.6x10^{-6}$	4038.8
Directional Gyro	$247.3x10^{-6}$	4048.6
Global Positioning System (GPS)	$85.7x10^{-6}$	11682
TAD Inhibit switch	$6.37x10^{-6}$	156985.9
TA Display – Weather Radar PPI	$227.1x10^{-6}$	4403.3
EGPWC	$80x10^{-6}$	125000

5.7.2 **Failure rates from field data**

The EGPWS failure rate field data has been obtained from airlines. The datasets were obtained from two different data sources in a form of spread sheet which contains removal events between 1st of January 2010 to 31st December 2010 and another spread sheet from January 2008 to December 2010. For each EGPWS removal event, the information obtained is as follows:

Dataset 1

- Part number
- Serial number
- Date of removal
- Reason for removal
- Aircraft registration number
- Vendor

Data set 2

- Aircraft type
- Ata chapter
- Part number
- Time since new
- Time since fault
- Time since overhaul
- Removal date
- Workshop note

5.8 EGPWS FAR regulation compliance

Table 6-2 presents the FAR regulation compliance with regards to EGPWS. It describes the applicable regulation and accepted probability of failure in possible failure condition of an EGPWS system.

Table 5-2: FAR regulation compliance

Failure condition	Applicable regulations	Probability of Failure P(F)	Section
Loss of all EGPWS Function	AC-23-18, 7.d.(2) (a)	8.031×10^{-5}	3.3
False Annunciation of Mode 1 "Pull Up" Warning	FAR, Part 23, 23.1309 (b) AC 23.1309-1C, 9.d.	7.375×10^{-6}	4.4
Unannunciated loss of the Mode 1 "Pull Up" Warning	FAR, Part 23, 23.1309 (b) AC 23.1309-1C, 9.d.	9.368×10^{-6}	5.5
False Annunciation of Mode 2 "Pull Up" Warning	FAR, Part 23, 23.1309 (b) AC 23.1309-1C, 9.d.	5.909×10^{-6}	6.5
Unannunciated loss of the Mode 1 "Pull Up" Warning	FAR, Part 23, 23.1309 (b) AC 23.1309-1C, 9.d. AC 23-18, 7.d.(2)(b)	6.783×10^{-6}	7.7
False Annunciation of Terrain Awareness "Pull Up" Warning	FAR, Part 23, 23.1309 (b) AC 23.1309-1C, 9.d. AC 23-18, 7.d.(2)(b)	1.669×10^{-6}	8.6
Unannunciated loss of the Terrain Awareness "Pull Up" Warning	FAR, Part 23, 23.1309 (b) AC 23.1309-1C, 9.d. AC 23-18, 7.d.(2)(c)	7.289×10^{-6}	9.9
Hazardously Misleading Information on the Terrain Awareness Display	AC 23-18, 7.d.(2)(d)	1.203×10^{-5}	10.9
Failure of the installed TAWS should not degrade any integrity of any installed system with the TAWS interfaces that could have either hazardous or catastrophic failure conditions as defines by AC23.1309-1C	AC 23-18, 7.d.(2)(e)	Qualitative analysis	1.5

5.9 SLAAP for EGPWS

Most current avionic systems utilize a federated architecture. Each line replaceable unit (LRU) is an independent device made by different manufacturers using potentially very different design approaches. Technological diversity and fractal design present a host of challenges to the avionics maintenance and logistics process. Each LRU manufacturers provide independent diagnostic capability for its unit in the form of built -in -test (BIT), automated test equipment (ATE), and test program sets (TPS).

Non-uniformity of test equipment and unrealized overlap of functional capability results in excess test resources at all levels of the maintenance system and inhibits interoperability through the inflexibility of process. Commonly lost in this process is the working requirement that these distinct avionic components function side by side, in a largely autonomous fashion, to provide the total system functionality required to fulfil the aircraft’s mission.

It is this integration and its potential system-level effects that have not been considered by the current maintenance infrastructure. Exposing this integration of avionics components through the capture and meaningful retention of all available data can contribute significant intelligence to avionics diagnostics and repair (Kalgren et. al., 2004).

SLAAP includes several models of computation such as the Markov Model, Kaplan Meier Chart, MTTF and Cox’s Regression Analysis. Different models of computation can be chosen with regards to availability of data to be analysed. SLAAP uses a standard graphical user interface with specific functions to be determined. The GUI is presented with a design window to insert inputs and calculate outputs or produce graphs.This behaviour is implemented in a specially formulated Matlab code.

CHAPTER 6

RESULTS

In this chapter, the application of methodology developed will be illustrated. Terrain Awareness Warning System (TAWS) was chosen as a test model for the analysis for several reasons. The first reason it is chosen because it is one of the equipment of avionics used on board aircraft and secondly, because prognostics are feasible and cost-worthy to be applied to safety related equipment for cost effectiveness. The other reasons are that the data needed for analysis is readily available online and thus the MTBUR and MTBF can be compared theoretically and thus, possible to validate field data obtained. The last reason is that the equipment falls under 'navigation' section of avionics that proves to have high breakdown rate.

6.1 Case studies for trend analysis

The sample data sets for the case studies were gathered from several airlines including Malaysian Airlines and Royal Brunei Airlines. These data consists of different types of component removal considered as discrepancies that was classified under several ATA chapters. Aircraft fleet involved as sampling consists of ATR 72-500, B727, B737, B737-400 (B734), Boeing 767-33AER, A319, and A320. There are also sample data that was specifically on EGPWS LRUs only. The objective of this section is to analyse and understand common problems and trends in maintenance line and then use the methods described in the previous chapters to establish results.

6.1.1 **Source 1 (ATR 72-500 component removal data)**

This data was gathered from Maswings and Firefly aircraft, companies which are both under Malaysian Airlines Berhad, based in Malaysia. Components removals were gathered between August 2008 and March 2010. The summary of the removals are presented in Table 6-1 and will be explained in the subsections that follow.

Table 6-1: Summary of ATR aircraft components removal

Airline	Date	Number of data
Maswings 1 (MW1)	22/9/2008-31/12/2009	416
Maswings 2 (MW2)	4/2/2010-1/3/2010	25
Firefly 1 (FF1)	22/8/2008-22/1/2010	510
Firefly 2 (FF2)	12/1/2010-8/3/2010	35

6.1.1.1 Maswings 1 data collection

For the Maswings 1 collection of data, the top three ATA chapters that produce highest component removal are from ATA 32 – Landing Gear (46%), ATA 24 – Electrical Power (11%) and ATA 34 – Navigation (6%) which occurs from 2008 to 2009 as shown in Figure 6-1. The removal was 86% unscheduled which is shown in Figure 6-3.

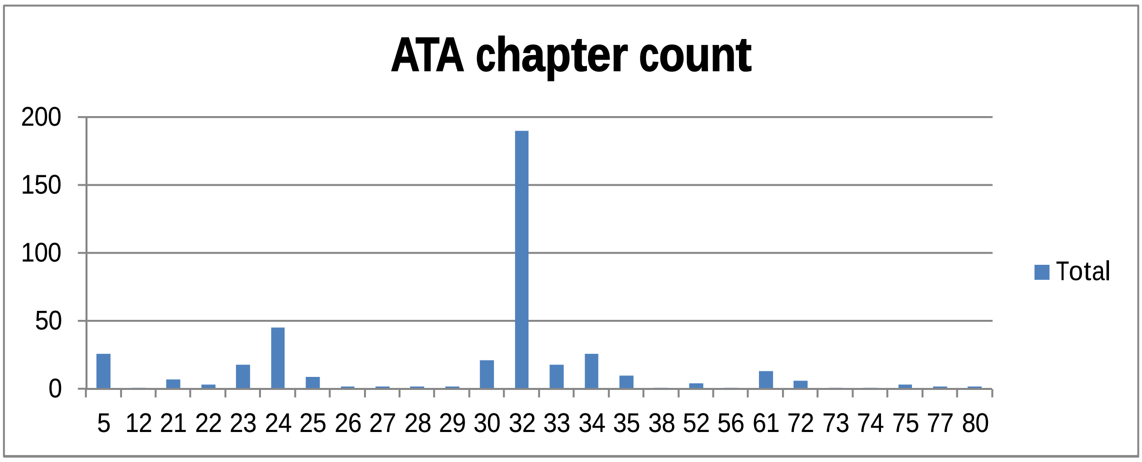

Figure 6-1: MW1 record of ATA chapter count

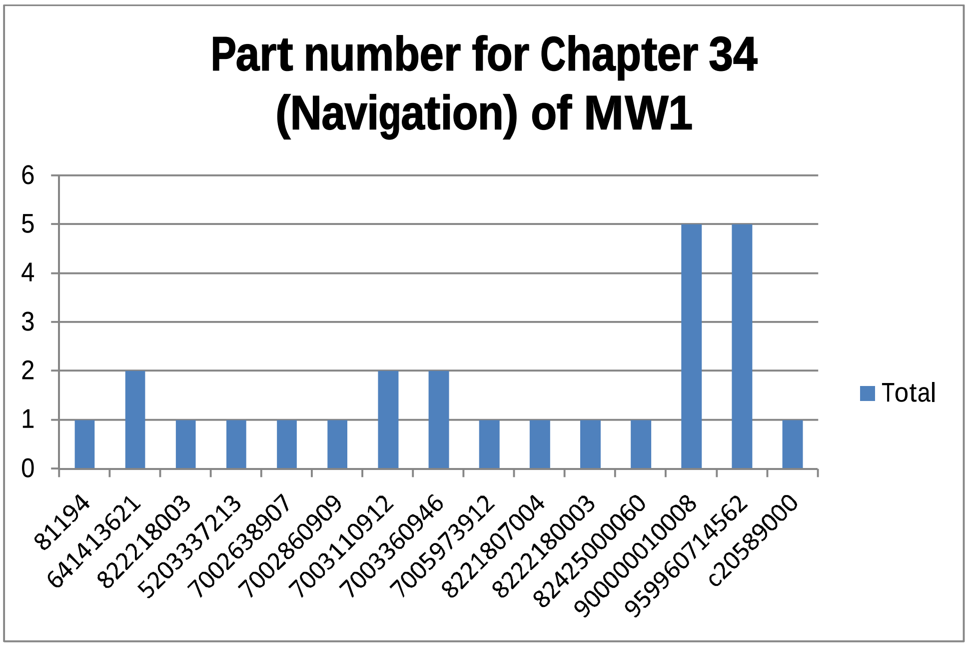

Figure 6-2: Common removal for Chapter 34 (Navigation)

Figure 6-2 shows the detail of data from ATA Chapter 34 of ATR aircraft data for MW1 whereby the common removal is from Terrain and Traffic Collision Avoidance System (T2CAS) labelled as 9000000-10008 and radio altimeter labelled as 959960714562. Both of these components are categorised as one of the components in TAWS or the position and warning system of aircraft.

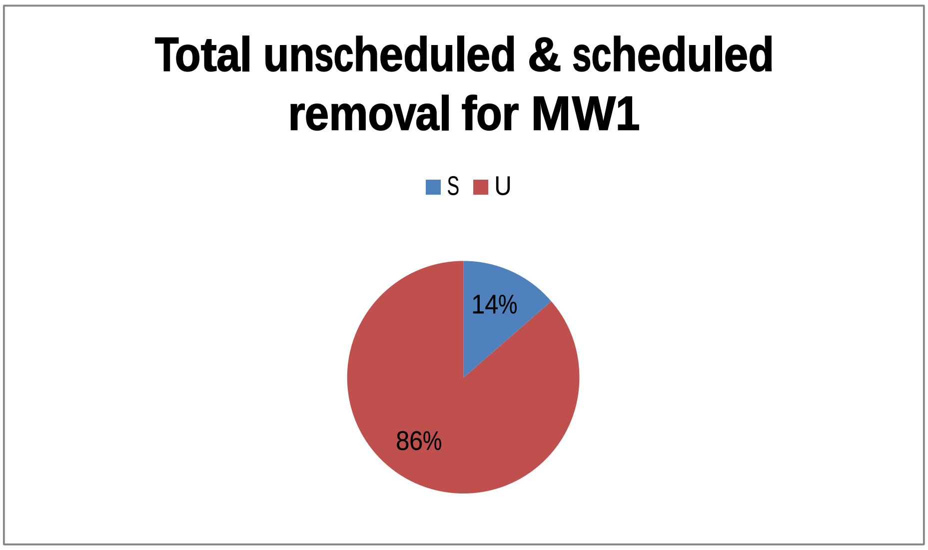

Figure 6-3: Scheduled versus Unscheduled removal of components of MW1

Figure 6-4 elaborates the number of Maswings 1 aircraft by category. Out of the 6 aircraft, 9M-MWA (MWA) has the highest removal with a count of components removal.

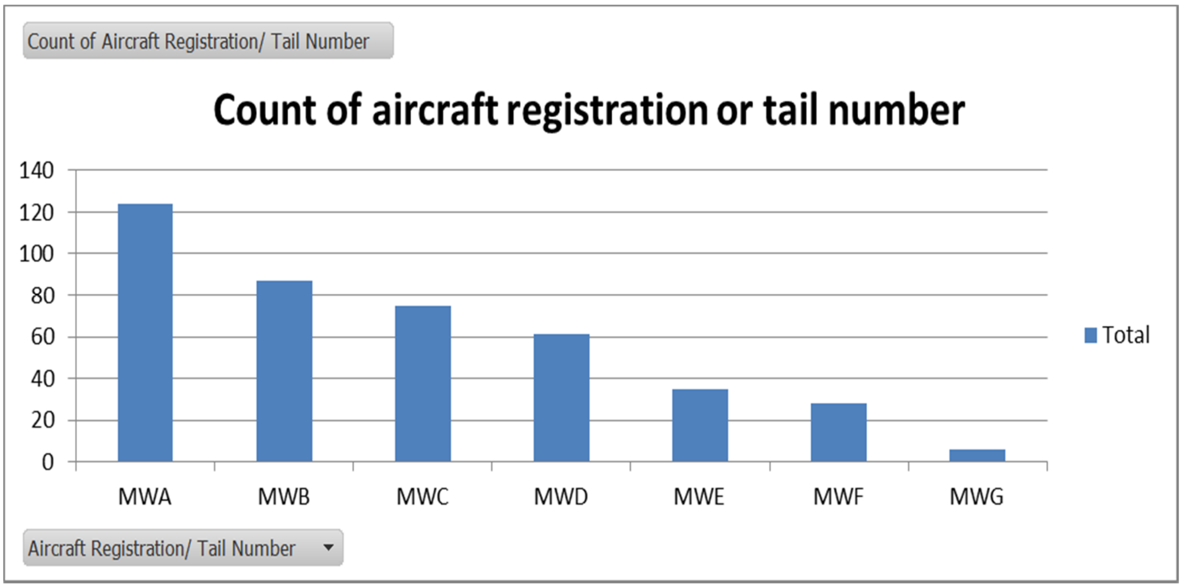

Figure 6-4: Count of aircraft registration or tail number of MW1

6.1.1.2 Maswings 2 data collection

From this record labelled as MW2, a number of 25 data was collected from ATA Chapters 23, 24, 30, 32, 34. The highest count of removal has been from ATA 32, which is the landing system. However, ATA 34, which is navigation, is among the highest three removals. Figure 6-5 shows the graph that illustrates the count for ATA chapters in MW2 record and Figure 6-6 shows the count of aircraft based on the registration number of category MW2.

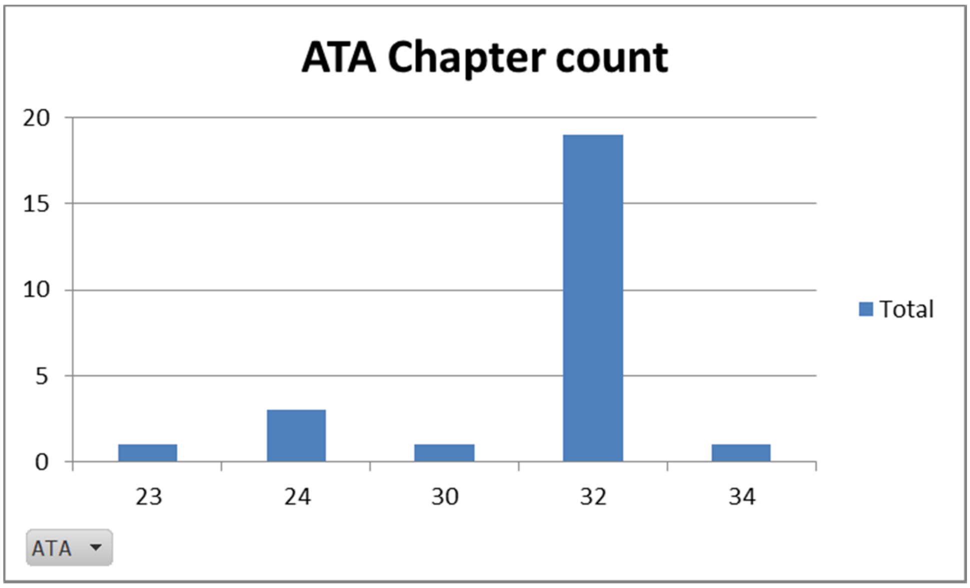

Figure 6-5: MW2 record of ATA chapter count

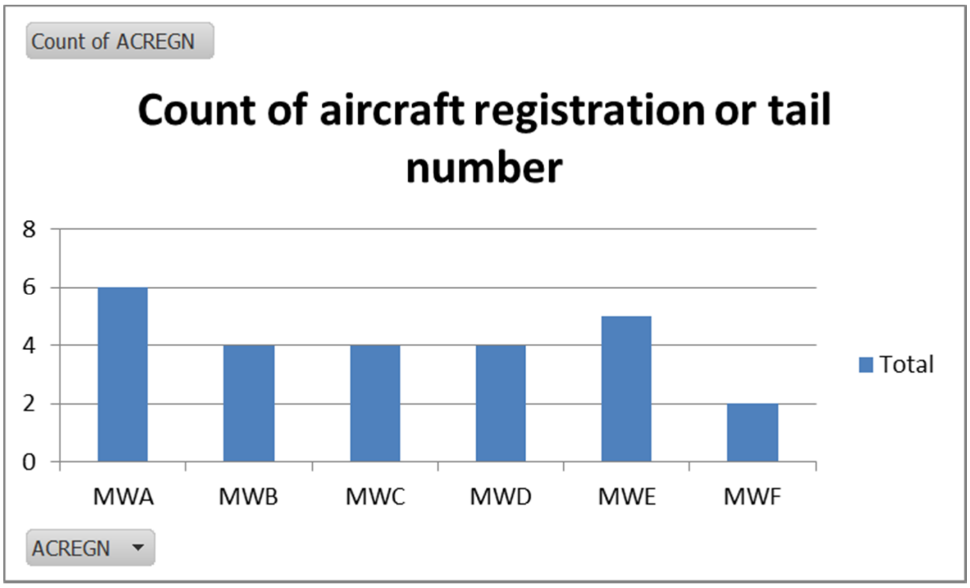

Figure 6-6: Count of aircraft registration or tail number of MW2

6.1.1.3 Firefly 1 data collection

This record gathered an amount of 510 samples of data. Based on Figure 6-7, the majority of removal is from ATA chapter 32 (landing gear). The Figure 6-8 shows once again the unscheduled removal being the majority cases as compared to a scheduled removal of components in an airlines maintenance line. Lastly, Figure 6-9 shows the count of aicraft registration or tail number for FF1 data.

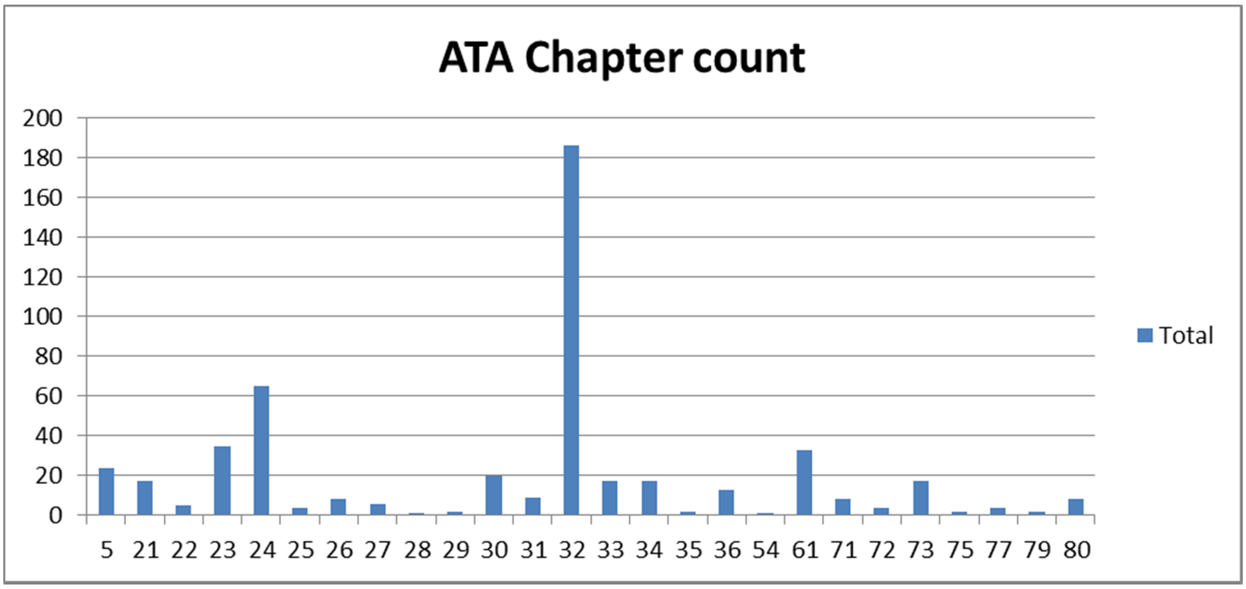

Figure 6-7: FF1 record of ATA chapter count

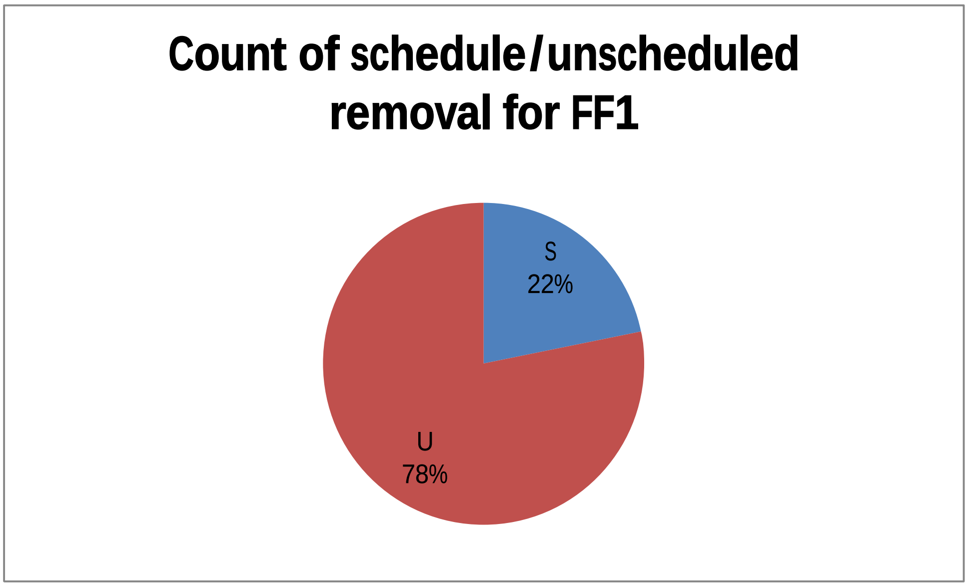

Figure 6-8: Scheduled versus Unscheduled removal of components of FF1

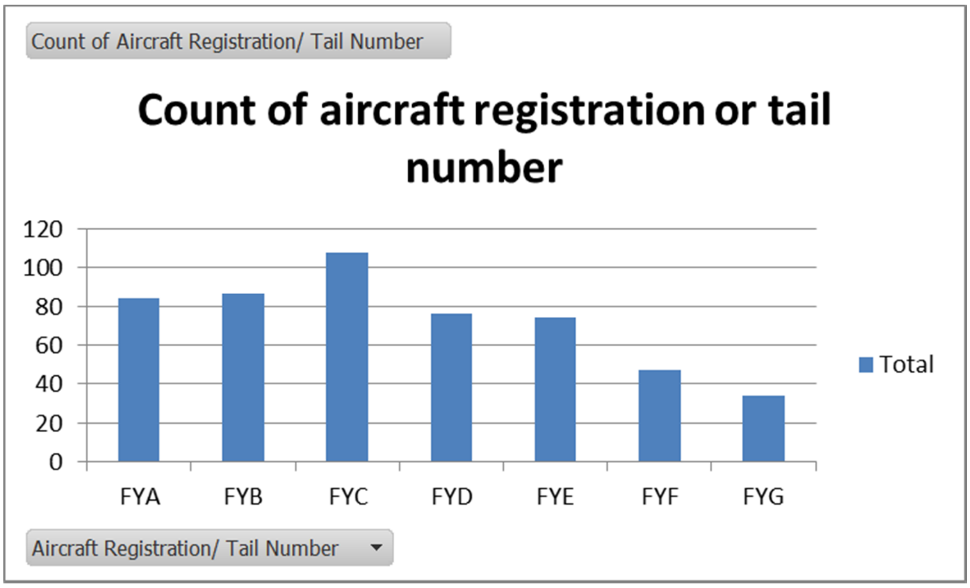

Figure 6-9: Count of aircraft registration or tail number of FF1

6.1.1.4 Firefly 2 data collection

From this record labelled as FF2, a number of 35 data was collected from ATA Chapters 24, 26, 27, 32, 36, 61. The highest count of removal has been from ATA 32, which is the landing system as shown in Figure 6-10 and count of aircraft registration or tail number of FF2 is shown in Figure 6-11.

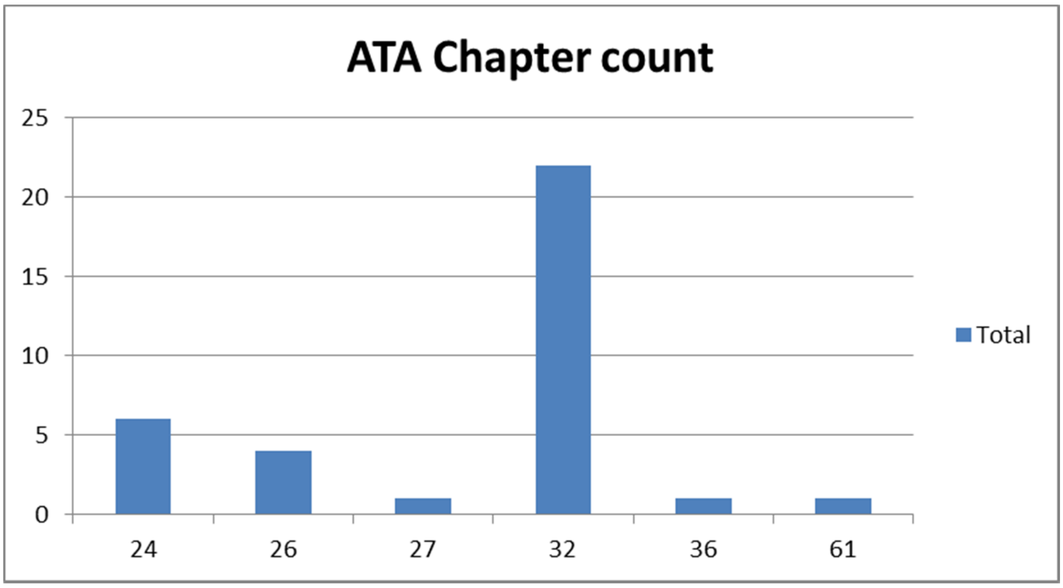

Figure 6-10: FF2 record of ATA chapter count

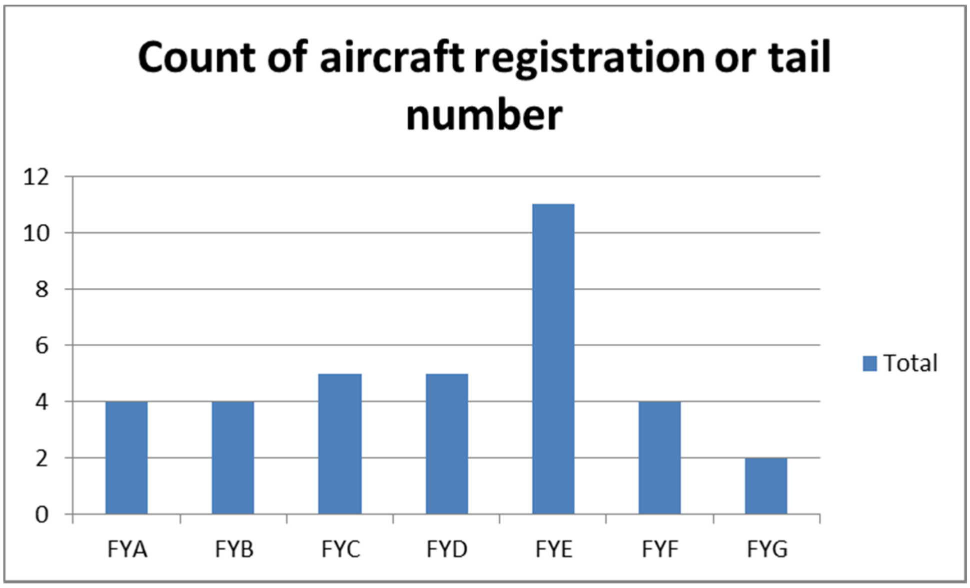

Figure 6-11: Count of aircraft or tail number of FF2

6.1.2 **Source 2 (EGPWS removal data for Boeing 727, 737)**

Table 6-2: EGPWS removal including EGPWC and Terrain Display Unit (TDU) for EGPWC and TDU

P/N	S/N	Removal date	Reason for removal	Aircraft Re	Vendor	Repair shop findings	Aircraft ty
80-5145-9-3	111	30/05/2008	FO EGPWS INOP	9M-TGG	HONEYWE	Nil record from Tspares	B727-247
80-5145-9-3	325	04/06/2008	CAPT EGPWS INSERVICEABLE	9M-TGG	HONEYWE	Defect confirmed	B727-247
80-5145-9-3	289	17/06/2008	CAPT TDU INOP	9M-TGH	HONEYWE	Defect confirmed	B727-247
80-5145-9-3	125	17/12/2008	FO TDU INOP	HS-SCH (T	HONEYWE	Defect not confirmed	B727-247
80-5145-9-3	327	22/06/2009	CAPT TDU INOP	9M-TGM	HONEYWE	Defect not confirmed	B727-200
80-5145-9-3	386	10/04/2010	CAPT TDU NIL DISPLAY	9M-TGG	HONEYWE	Defect confirmed	B727-247
80-5145-9-3	323	14/06/2010	FO TDU GOES BLANK AND HOT	9M-TGB	HONEYWE	Defect not confirmed	B727-200
80-5145-9-3	111	02/10/2010	FO TDU INOP	9M-TGM	HONEYWE	Defect not confirmed	B727-200
80-5145-9-3	288	15/11/2010	CAPT TDU GOES BLANK	9M-TGB	HONEYWE	Defect confirmed	B727-200
80-5145-9-3	323	29/12/2010	CAPT TDU UNABLE TO ADJ BRIGHTNESS	9M-TGB	HONEYWE	Defect not confirmed	B727-200
80-5145-9-3	199	30/12/2008	CAPT TDU UNSERVICEABLE	9M-PMW	HONEYWE	Defect confirmed	B737
80-5145-9-3	9570	27/06/2010	CAPT TDU GOES BLANK	9M-PMW	HONEYWE	Defect confirmed	B737
965-1076-020-212-212	2707	28/03/2008	Both captain and FO terrain warning INOP	HS-SCJ (T	HONEYWE	Nil record from Tspares	B727-247
965-1076-020-212-212	487	04/08/2008	No terrain displayed on TDUs	9M-TGE	HONEYWE	Defect confirmed	B727-200
965-1076-020-212-212	N/A	22/09/2008	FO TDU INOP	9M-TGM	HONEYWE	Defect confirmed	B727-200
965-1076-020-212-212	2707	17/11/2008	GPWS MODE "TERRAIN" INOP	HS-SCJ (T	HONEYWE	Defect not confirmed	B727-247
965-1076-020-212-212	3791	29/05/2009	EGPWS INOP	9M-TGH	HONEYWE	Defect not confirmed	B727-247
965-1076-020-214-214	2876	26/11/2009	WINDSHEAR AND GPWS FAIL LIGHTS ON AND "NO TERRAIN" & TERRAIN SYSTEM OVERRIDE LIGHTS REMAINS ON	9M-TGG	HONEYWE	Defect confirmed	B727-247
965-1076-020-212-212	4123	26/03/2010	TERRAIN SYSTEM OVERRIDE LIGHTS REMAIN ON	9M-TGG	HONEYWE	Defect not confirmed	B727-247
965-1076-020-212-212	3747	04/06/2010	TERRAIN WARNING SYSTEM INOP	9M-TGE	HONEYWE	Defect not confirmed	B727-200
965-1076-020-212-214	3270	11/07/2010	TERRAIN WARNING DISPLAY INOP	9M-TGE	HONEYWE	Defect confirmed	B727-200
965-1076-020-212-214	3270	13/05/2010	EGPWS NUISSANCE "PULL UP" WARNING CAME ON WHEN ESTABLISHED ON GLIDEPATH	9M-PML	HONEYWE	Defect not confirmed	B737
965-1076-020-212-214	2876	18/07/2010	EGPWS "NO TERRAIN" DISPLAYED ALL THE TIME	9M-PML	HONEYWE	Defect confirmed	B737
965-1076-020-212-212	3747	11/07/2010	EGPWS "NO TERRAIN" DISPLAYED ALL THE TIME	9M-PML	HONEYWE	Defect confirmed	B737

Table 6-2 describes the data for EGPWS removal from 30th May 2008 to 29th of December 2010. These data were gathered from Transmile Airlines now known as Raya Airways.

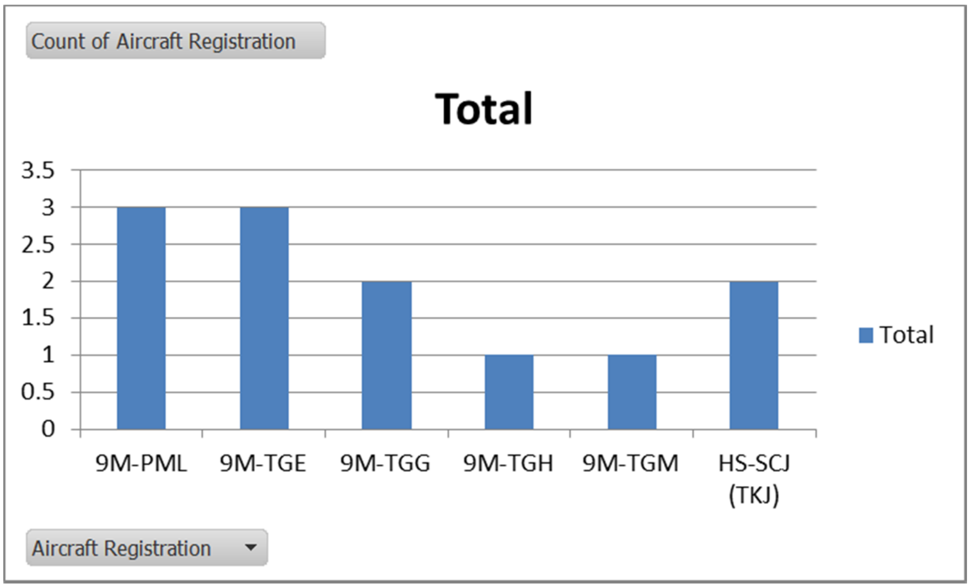

Figure 6-12: Count of aircraft registration for EGPWC removal

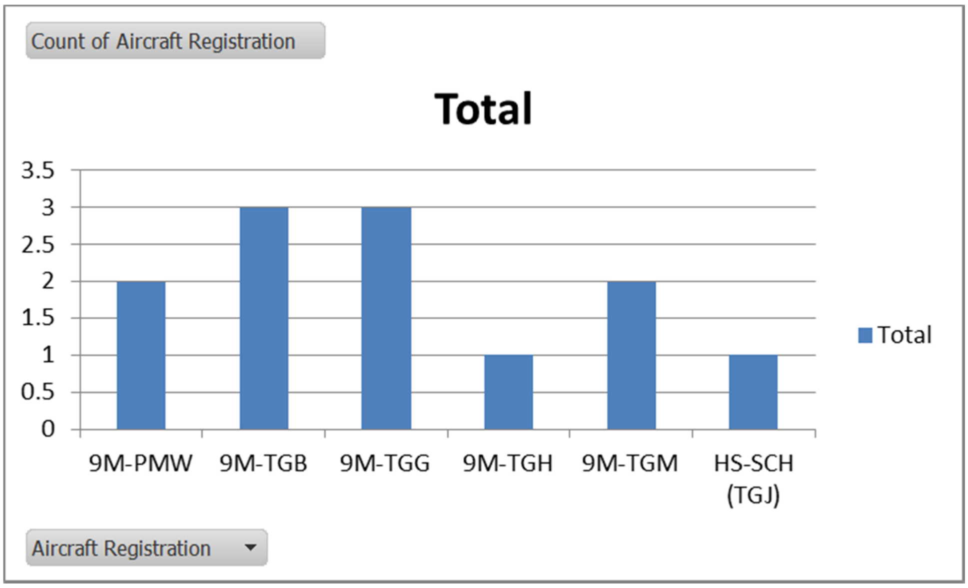

Figure 6-13: Count of aircraft registration for TDU removal

The data in Table 6-2 consists of EGPWC and TDU removal of aircrafts of B727 and B737. Figure 6-12 and Figure 6-13 show the summary of aircraft count based on aircraft registration for the removal of EGPWC and TDU. The total unscheduled removal over period of 3 years for EGPWC is 12 and the total unscheduled removal over period of 3 years for TDU is also 12. Based on fleet, for EGPWC, B727 contributes a total of 9 removals while B737 contributes a total of 3 removals. For TDU removals, a number 10 removal has been made on B727 and 2 removals have been done on B737. The Table 6-3 illustrates the number of flight hours for the two aircraft in 2008, 2009 and 2010. Table 6-4 shows the total flight hours characterised by type of aircraft for the three consecutive years.

Table 6-3: Yearly flight hours recorded for B727 and B737

	2008		2009		2010	
Flight hours	B727	B737	B727	B737	B727	B737
	8790	2583	8350	675	7729	1667
Total Flight hours	11373		9025		9396	

Table 6-4: Total flight hours for the three years for B727 & B737

B727 Total Flight Hours (2008 through 2010)	8790+8350+7729	24869
B737 Total Flight Hours (2008 through 2010)	2583+675+1667	4925

With the removal records from the sample data, the MTBUR and MTBF can be calculated. From the MTBF, failure rate can then be known. With failure rate value known, it can then be compared with the benchmark given by the original equipment manufacturer (OEM). The OEM for this product is Honeywell International Incorporated.

In order to determine the MTBUR *Equation* 6-1 and *Equation* 6-2 has been used:

$$MTBUR = \frac{flight\_hours \times units\_installed\_per\_aircraft}{number\_of\_unscheduled\_removal\_for\_that\_period}$$

(Equation 6-1)

And, in order to calculate the MTBF, this equation has been used:

$$MTBF = \frac{flight\_hours \times units\_installed\_per\_aircraft}{number\_of\_confirmed\_failure\_during\_that\_period}$$

(Equation 6-2)

From Honeywell Product specification which is available online as characterised according to function and according to component in the fault tree diagram; the loss of all EGPWS functions given the probability per flight hour is presented to be 8.031×10^{-5} and so, the calculated MTBF is 12451.7 hours. The given the failure rate of EGPWC per flight hour is 80×10^{-6}, so, the MTBF is then 12500 hours.

6.1.2.1 EGPWC removal

The calculation for MTBUR and MTBF is shown below using equations 7-1 and 7-2.

<u>MTBUR for EGPWC:</u>

B727 MTBUR (EGPWC) = $24869 \times \frac{1}{9} = 2763.3\ hours$

B737 MTBUR (EGPWC) = $4925 \times \frac{1}{3} = 1642\ hours$

Thus, the average MTBUR (EGPWC) is 2202.5 hours

<u>MTBF for EGPWC:</u>

B727 MTBF (EGPWC) = $24869 \times \frac{1}{4} = 6217.25\ hours$

B737 MTBF (EGPWC) = $4925 \times \frac{1}{2} = 2462.5\ hours$

Average MTBF (EGPWC) = 4339.9 hours

Thus, failure rate for EGPWC is then 230.42×10^{-6} per hour

6.1.2.2 TDU removal

The calculation for MTBUR and MTBF is shown below using equations 7-1 and 7-2.

<u>MTBUR for TDU:</u>
B727 MTBUR (TDU) = $24869 \times \frac{2}{10} = 4974\ hours$
B737 MTBUR (TDU) = $4925 \times \frac{2}{2} = 4925\ hours$
Average MTBUR (TDU) = 4949.5 hours

<u>MTBF for TDU:</u>
B727 MTBF (TDU) = $24869 \times \frac{2}{4} = 12434.5\ hours$
B737 MTBF (TDU) = $4925 \times \frac{2}{2} = 9850\ hours$
Average MTBF (TDU) = 111142 hours

Thus, the average MTBF (EGPWC and TDU) is 57741 hours,

It can be concluded that the failure rate for the items is as above expected performance standard whereby the MTBF benchmark for EGPWS was found to be 12451.7 hours.

6.1.3 **Source 3 (EGPWS removal data 737-400)**

In this set of TAWS/EGPWS removal data of Boeing 737-400 aircraft, a number of 35 samples have been collected. Figure 6-14 shows the monthly removal trend for the particular aircraft. However, only 16 (46%) was a confirmed failure and 19 (54%) has been labelled as *defect not confirmed*. From the records, the MTBUR was calculated using *Equation (6-1)* and are found to be 2617.257 hours and the calculated MTBF using *Equation (6-2) to be* 7046.462 hours. The total flight hours recorded for the sample given is 91604 hours. So, from the MTBF number then, failure rate is found to be 141.915×10^{-6} hours.

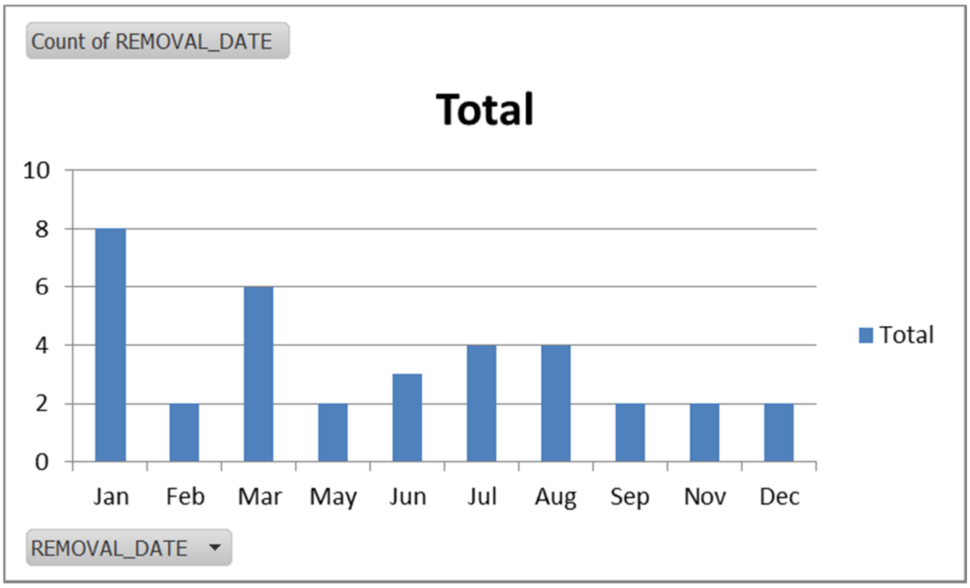

Figure 6-14: Removal of EGPWS monthly for B737-400 (B734)

With this set of data, Kaplan Meier chart can be used to calculate the time to failure which was found to be 40000 plus hours (42717) as shown in Figure 6-15.

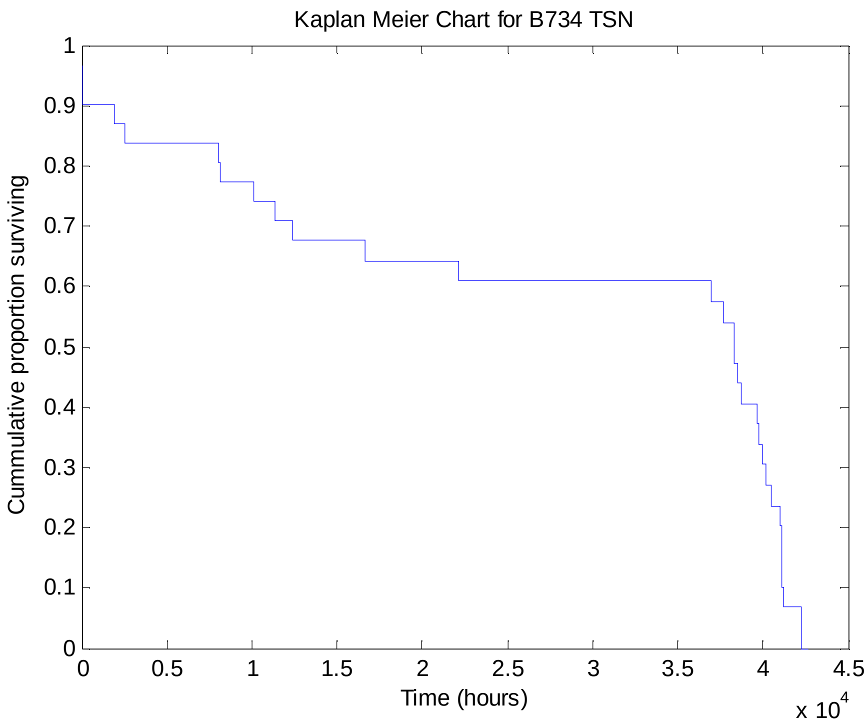

Figure 6-15: Kaplan Meier Chart for B737-400 (B734) using the TSN value

An example of a Matlab software based GUI for the Kaplan Meier simulation produced an output as shown in Figure 6-16 which was run using input given in the Appendix. Inputs include State Transition (ST), Number at risk, and Number of failure.

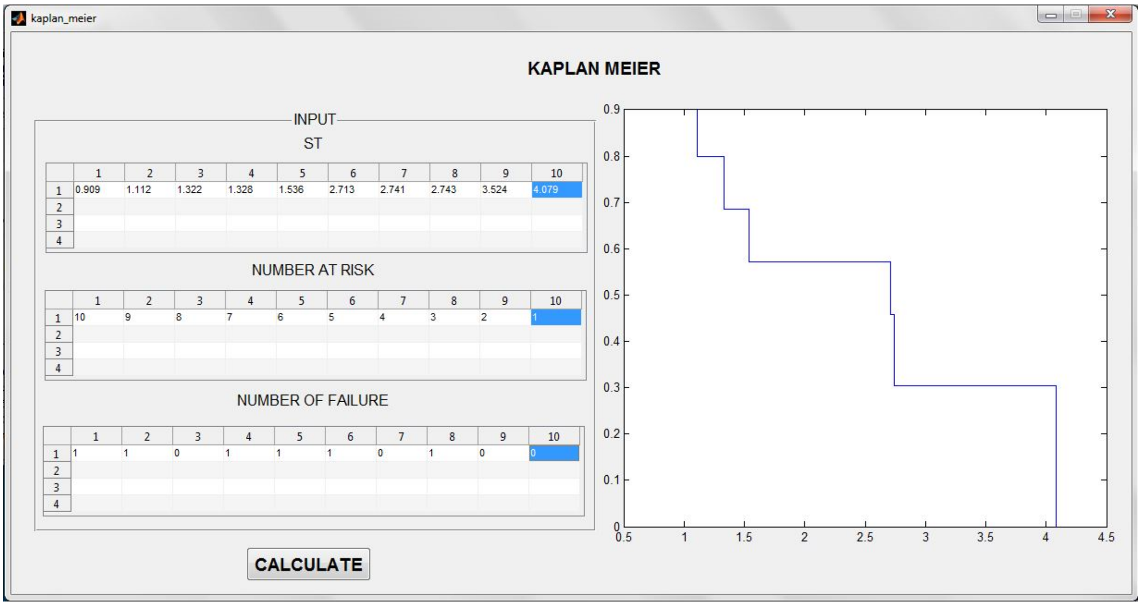

Figure 6-16: Kaplan Meier GUI

6.1.4 **Source 4 (ATA 34 removal data Airbus)**

This data contains removal report for Airbus A319 and Airbus A320 aircrafts for the year 2003 through 2010 from Royal Brunei Airlines. The data sample was specifically chosen for ATA chapter 34 which focused on Navigation Instruments. It has been labelled using A, B, C, and D and consists of the following number of samples:

1. Airbus A- 327 samples
2. Airbus B- 291 samples
3. Airbus C- 93 samples (up to 2006 only)
4. Airbus D- 334 samples

6.1.4.1 Airbus A

These data sets contain removal report from 26th August 2003 to 9th of January 2011. Out of these data sets, 13 reported on GPWS. One example of recorded report states ‘NAV GPWS FAULT DURING CLB, MSG DISAPPEARED ON

LANDING'. From Table , MA refers to Maintenance Report and PI refers to defects reported by pilots and it can be concluded that faults were detected during flight or during pre-flight as faults were detected and reported by pilots.

Table 6-5: Summary of removal report for ATA 34

Airbus	Dates	Sample Count	MA	PI
A	26/8/2011-9/1/2011	327	46%	54%
B	8/9/2003- 4/1/2011	291	53%	47%
C	16/1/2004-9/3/2006	93	39%	61%
D	25/9/2010-9/1/2011	334	32%	68%

6.1.5 **Source 5 (ATA 34 removal data Boeing)**

The data gathered for this section was from Royal Brunei Airlines. The aircraft under analysis were six Boeing 767-33AER aircraft. These data includes discrepancies dated from 2nd January 2009 to 15th December 2010. The data consisted of 523 recorded discrepancies which falls under ATA 34. All categories under the ATA 34 such as the 3410, 3420 and 3460 were highlighted as shown in Figure 6-17.

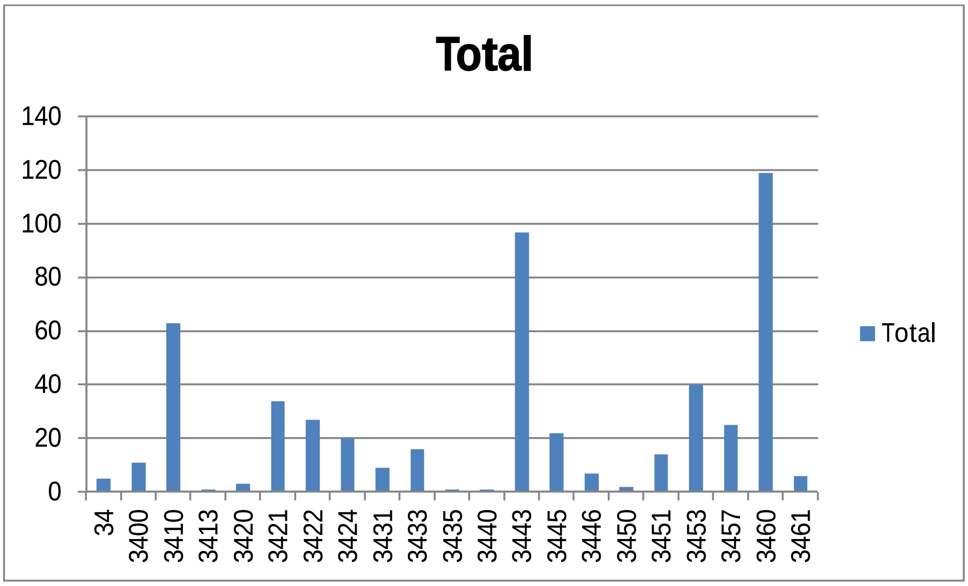

Figure 6-17: A two-year discrepancies of ATA 34 of B767 aircraft

6.2 Case study for Proportional Hazard Ratio

A series of analysis was done to calculate the beta values of covariates (temperature and stress) using the failure rates generated for a digital circuit board. The purpose of this analysis is to study the effect of temperature and stress on failure rates of the device. In this case, the failure rate is fitted for Cox Proportional hazard function with the variable, X being temperature and stress using some sample data in Table 7-6. There are two environment classifications to the data which are the ground benign and ground fixed. The table lists the failure rates at different combination of temperature and stress conditions of a digital circuit board. An assessment of the significance of the predictor variable will follow afterwards.

Table 6-6: Digital circuit board failure rates in 10^6 part-hours (Denson, 1998)

	Ground benign				Ground fixed			
Temperature	10°C		70°C		10°C		70°C	
Stress	10%	50%	10%	50%	10%	50%	10%	50%
ALCATEL	6.59	10.18	13.30	19.89	22.08	29.79	32.51	47.27
Bellcore Issue 4	5.72	7.09	31.64	35.43	8.56	10.63	47.46	53.14
Bellcore Issue 5	8.47	9.25	134.45	137.85	16.94	18.49	268.90	275.70
British Telecom HDR4	6.72	6.72	6.72	6.72	9.84	9.84	9.84	9.84
British Telecom HDR5	2.59	2.59	2.59	2.59	2.59	2.59	2.59	2.59
MH-217E Notice 1	10.92	20.20	94.37	111.36	36.38	56.04	128.98	165.91
MH-217F Notice 1	9.32	18.38	20.15	35.40	28.31	48.78	45.44	79.46
MH-217F Notice 2	6.41	9.83	18.31	26.76	24.74	40.15	73.63	119.21

To compare empirical methodologies, the failure rates in Table 6-6 were each calculated for each combination of environment. The analysis using Cox's hazard function is represented in Table 6-7. Using the beta values found, prediction of hazard ratio between different environments can be analysed.

Table 6-7: Results of "covariate b" or the coefficient, b using Cox's Regression analysis for temperature and stress on failure rates

Ground benign								Ground fixed							
Temperature constant				Stress constant				Temperature constant				Stress constant			
10°		70°		10%		50%		10°		70°		10%		50%	
10% 0	70% 1	10% 0	50% 1	10° 0	70° 1	10° 0	70° 1	10% 0	70% 1	10% 0	50% 1	10° 0	70° 1	10° 0	70° 1
0.3034		0.2721		0.6648		0.5054		-0.8043		-0.3615		-1.6139		-1.2648	

These results from Table 6-7 are interpreted using the GUI as shown in Figure 6.18 and Figure 6.19.

For example, if Ground Fixed was considered at 10% constant stress, covariate b was calculated to be -1.6139. Hazard ratio in this case has also been calculated. The estimated hazard ration calculated using exp(b) is 0.1991. This carries a meaning that hazard for 70° is 0.1991 higher than that of 10° temperature. The explanation is based on the following:

To predict survival as a function of a dichotomous IV such as an Experimental v control groups;

In such cases, the IV is treated as a dummy variable and coded either 0 or 1

The resulting Cox regression model:

$$h(t) = [h_0(t)]\, e^{(b_1 X_1)}$$

When X = 0, $h(t) = [h_0(t)]\,(1)$, since $e^0 = 1$

When X = 1, $h(t) = [h_0(t)]\, e^{(b_1)(1)}$

b_1 = Cox regression coefficient, determined by partial likelihood estimation using matlab function

Linearizing the Hazard Function with a Dichotomous Independent Variable

$h(t) = [h_0(t)]\, e^{(b_1 X_1)}$

Dividing both sides by $h_0(t)$

$$\frac{h(t)}{h_0(t)} = \frac{[h_0(t)]\, e^{(b_1 X_1)}}{[h_0(t)]}$$

$$h(t) = \left(\frac{e^{(b_1 X_1)}}{h_0(t)} \right)$$

This is the hazard ratio or relative hazard which is Exp (b).

This ratio indicates the expected change in the risk of the terminal event when X changes from 0 to 1. (i.e. 1 = presence of the characteristic X)

When X = 0, the hazard ratio = 1.0

When X = 1, the hazard ratio Exp(b) = e **(b_1)**

<u>Possible Relationships</u>

If the hazard ratio = 1; The IV does not affect survival.

If the hazard ratio $<$ 1; The IV is associated with increased survival

If the hazard ratio is $>$ 1; The IV is associated with decreased survival

If the hazard ratio = 1; The parameter does not affect the time to failure.

If the hazard ratio $<$ 1; The parameter is associated with decreased time to failure

If the hazard ratio is $>$ 1; The parameter is associated with increased time to failure.

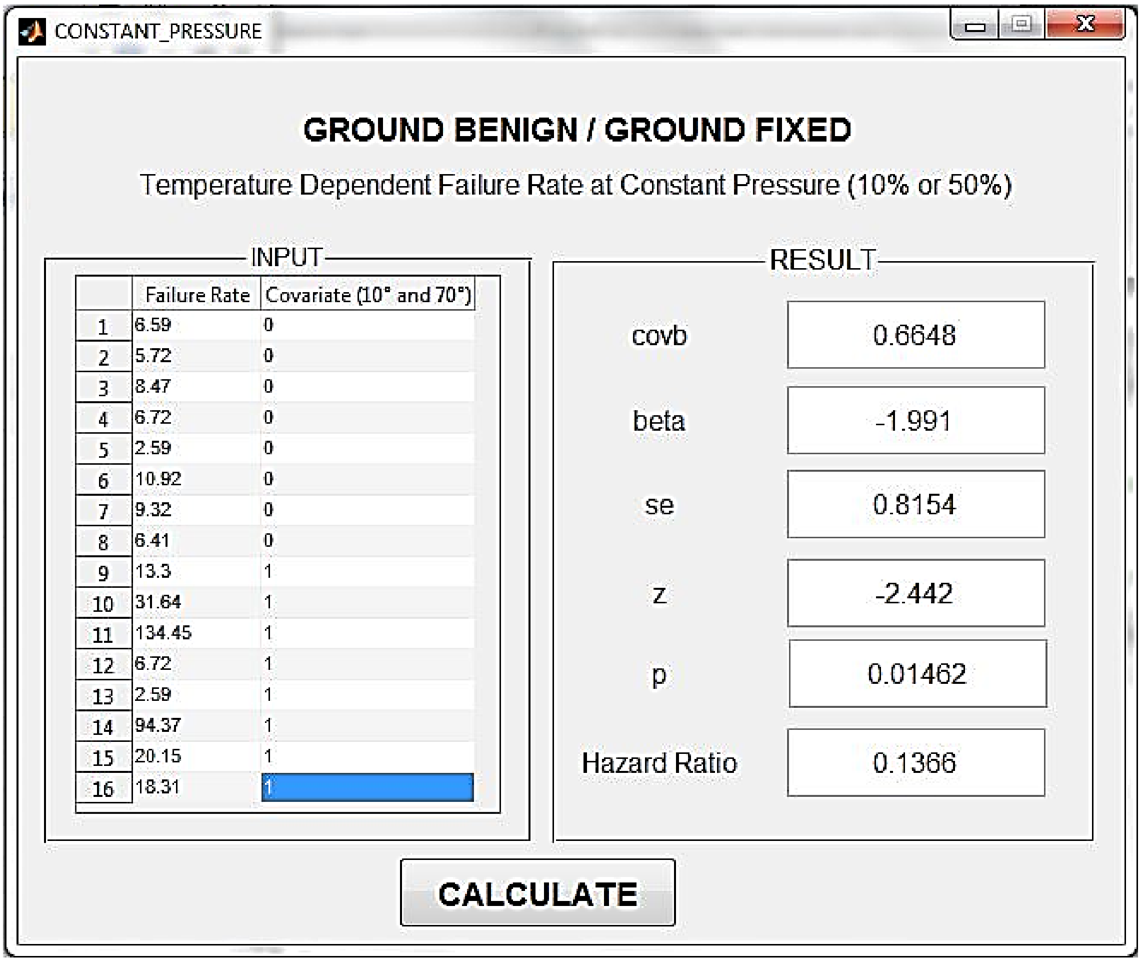

Figure 6-18: GUI for calculating the Hazard Ratio of Constant Pressure

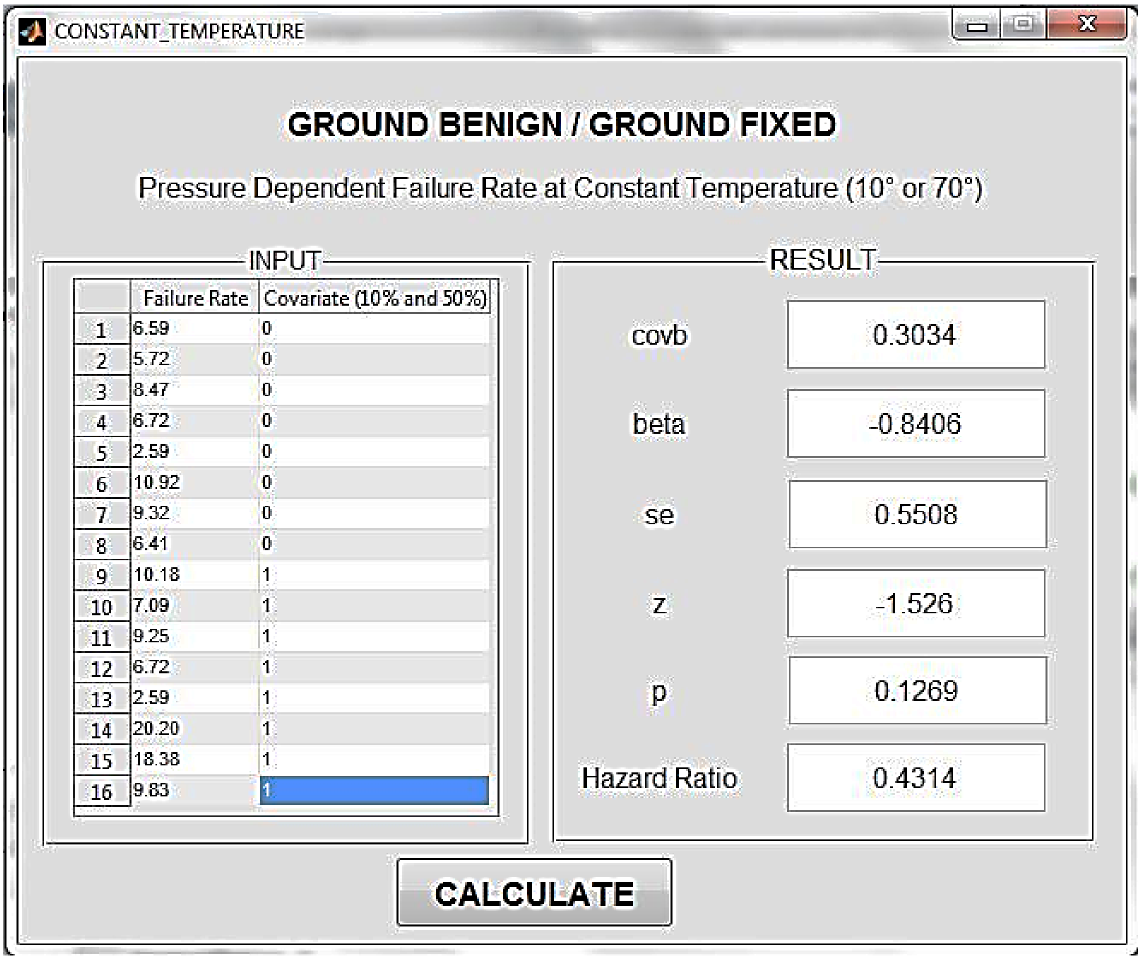

Figure 6-19: GUI for calculating the Hazard Ratio for Constant Temperature

6.3 Case study using Markov Model

The case study for Markov Model illustrates the use of fault tree to find the failure rates at different stages of a system. From the fault tree diagram below, mean time to failure (MTTF) can be calculated using Markov model. The diagram below has been simplified to show less complicated method in achieving time to failure value. This method uses eigenvalues and eigenvectors in finding estimated failure time. As such, any system with known fault tree diagram and failure rate of components can apply such method easily. As opposed to just fault tree diagram, Markov model gives a quantitative insight to a problem and will be an added advantage for top level, black box analysis for any system. In a way, Markov model uses linear regression analysis in solving P(t).

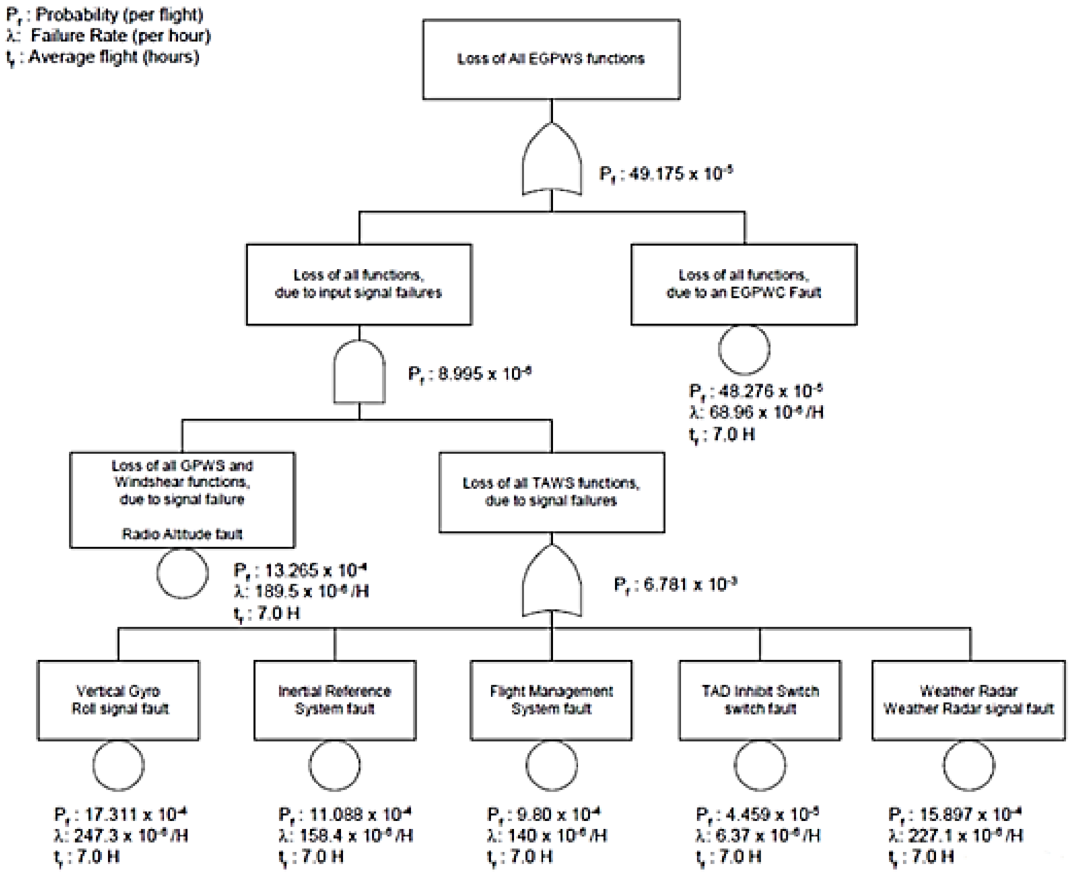

Figure 6-20: EGPWC fault tree from product specification list

From the fault tree in Figure 6-20, the minimum cut set of the above fault tree can be simplified. It only highlights the major events which affect loss of all EGPWS functions as shown in Figure 6-21. The simplified fault tree has been labelled appropriately with representation of A, B and C as the bottom level event as shown in Figure 6-22. From there, the conversion of fault tree to Markov Model has been made. With the failure rate value fitted in in the Markov Model, the links or state transition diagram is drawn for further analysis. This is shown in Figure 6-23, where the failure rates are fitted into the transition diagram which shows the event change for all possible states.

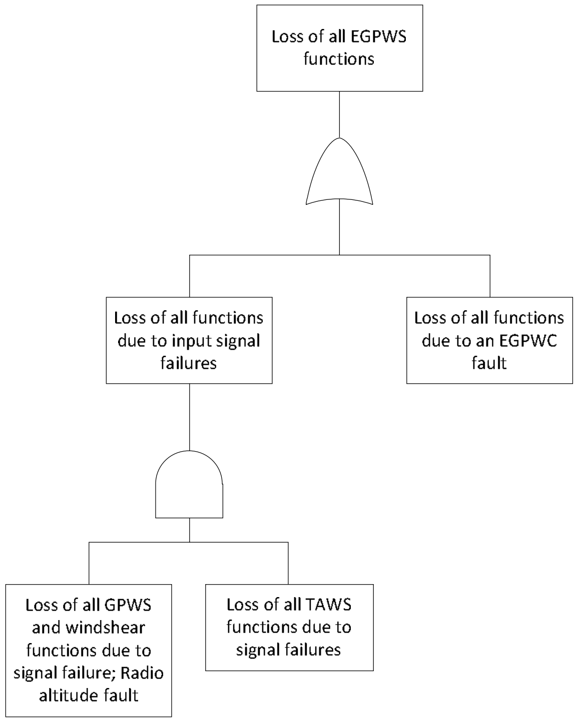

Figure 6-21: Simplified fault tree

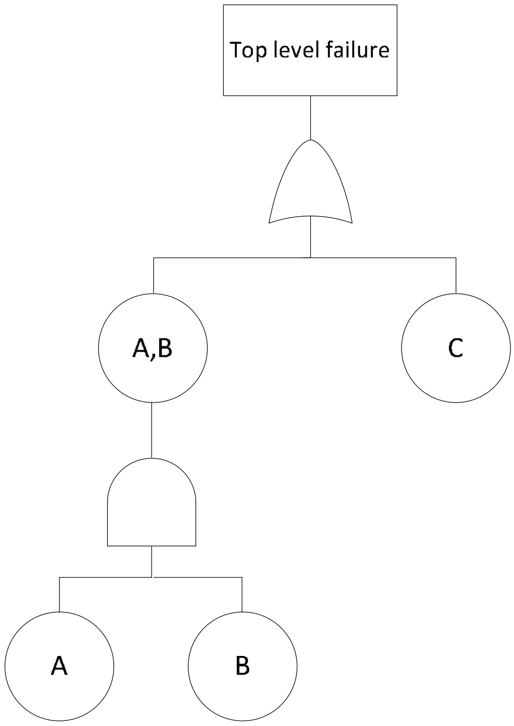

Figure 6-22: Basic fault tree from simplified diagram

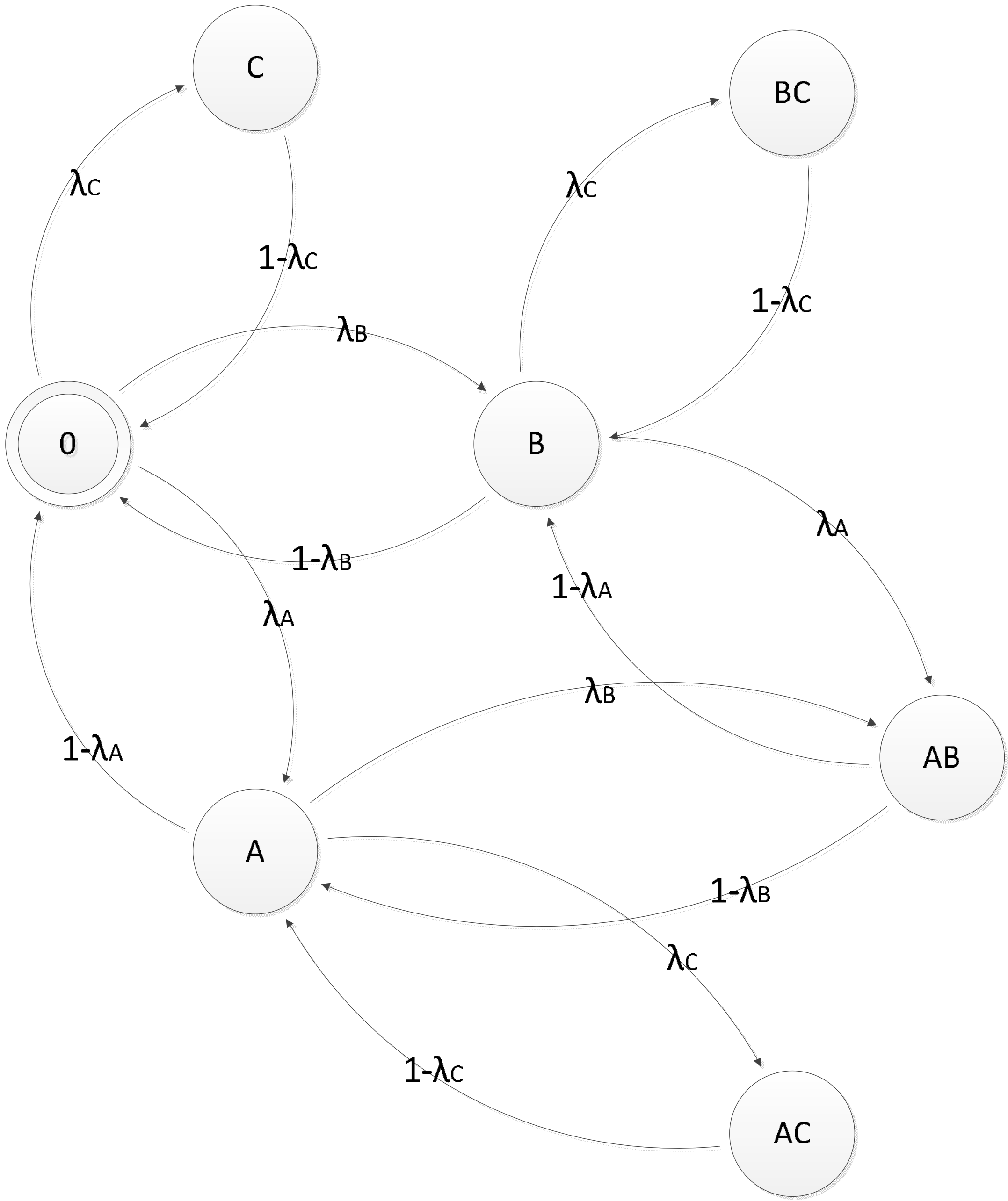

Figure 6-23: Markov state diagram for top level EGPWS fault tree

Failure rate value derived from the fault tree from product specification has allocated as in Table 6-8:

Table 6-8: Failure rate from fault tree

λ_A	0.0001895
λ_B	0.0009687
λ_C	0.000068966
$1-\lambda_A$	0.9998105
$1-\lambda_B$	0.9990313
$1-\lambda_C$	0.999931

Step 1: Set up transition matrix (Q) from failure rate data

Step 2: Find P(t) by finding eigenvalues and eigenvectors of Q

Step 3: Establish equation for P(t)

Step 4: Determine limiting distribution for each state

Step 5: Set up minimum cut set transition matrix to find $P_{new}(t)$

Step 6: Determine $R(t) = \sum P_{new}(t)$

Step 7: Find MTTF using $MTTF = \int_0^\infty R(t)dt$

The entire algorithm above has been realised using MATLAB and can be seen in Figure

```
Q=[-0.0012272 0.0001895 0.0009687 0 0.00006896 0 0;0.9998105 -1.0008 0
0.0009687 0 .000068966 0; 0.9990313 0 -0.9992899 189.5E-6 0 0
0.000068966; 0 0.9990313 0.9998105 -1.9988 0 0 0; 0.999931 0 0 0 -
1.0011 189.5E-6 0.9687E-3; 0 0.999931 0 0 0.9998105 -1.9997 0; 0 0
0.999931 0 0.9990313 0 -1.999]
[V D]=eig(Q)
syms t;
P0=[1 0 0 0 0 0 0];
P=P0*V*expm(D*t)*inv(V);
Pt=vpa(P)
```

Q =

-0.0012	0.0002	0.0010	0	0.0001	0	0
0.9998	-1.0008	0	0.0010	0	0.0001	0
0.9990	0	-0.9993	0.0002	0	0	0.0001
0	0.9990	0.9998	-1.9988	0	0	0
0.9999	0	0	0	-1.0011	0.0002	0.0010
0	0.9999	0	0	0.9998	-1.9997	0
0	0	0.9999	0	0.9990	0	-1.9990

V =

-0.3780	0.0000	0.0000	0.0000	0.0001	0.0006	-0.0000
-0.3780	-0.0002	-0.0000	-0.0007	-0.5780	-0.0003	0.0006
-0.3780	-0.0001	-0.0000	-0.0001	0.0003	-0.5808	0.0004
-0.3780	0.1528	0.1014	0.7069	-0.5778	-0.5817	0.0010
-0.3780	-0.0006	-0.0001	0.0007	0.0019	0.0120	0.5770
-0.3780	0.9058	-0.9437	-0.1211	-0.5762	0.0117	0.5776
-0.3780	0.3952	0.3148	-0.6969	0.0022	-0.5693	0.5775

D =

-0.0000	0	0	0	0	0	0
0	-2.0005	0	0	0	0	0
0	0	-1.9995	0	0	0	0
0	0	0	-1.9999	0	0	0
0	0	0	0	-1.0000	0	0
0	0	0	0	0	-1.0000	0
0	0	0	0	0	0	-1.0000

Pt =

0.99877*exp(0*t)+0.17258e-6*exp(-2 *t)+0.371e-7*exp(-1.999*t)+0.536e-7*exp(-1.999*t)+0.1883e-3*exp(-1*t)+0.9668e-3*exp(-1*t)+0.7148e-4*exp(-1*t),

0.1893e-3*exp(0*t)-0.1038e-6*exp(-2*t)-0.854e-8*exp(-1.999*t)-0.842e-7*exp(-1.999*t)-0.189e-3*exp(-1*t)-0.991e-7*exp(-1*t)-0.204e-6*exp(-1*t),

0.9685e-3*exp(0*t)-0.142e-6*exp(-2*t)-0.539e-7*exp(-1.999*t)-0.5464e-7*exp(-1.999*t)+0.467e-6*exp(-1*t)-0.967e-3*exp(-1*t)-0.131e-5*exp(-1*t),

0.1836e-6*exp(0*t)+0.731e-7*exp(-2*t)+0.2529e-7*exp(-1.999*t)+0.852-7*exp(-1.999*t)-0.1831e-6*exp(-1*t)-0.18363e-6*exp(-1*t)-0.4458e-9*exp(-1*t),

0.6888e-4*exp(0*t)-0.996e-7*exp(-2*t)-0.118e-7*exp(-1.999*t)+0.316e-7*exp(-1.999*t)+0.2069e-6*exp(-1*t)+0.878e-6*exp(-1*t)-0.6988e-4*exp(-1*t), 0.1306e-7*exp(0*t)+0.3081e-7*exp(-2*t)-0.1674e-7*exp(-1.999*t)-0.10382*exp(-1.999*t)-0.12986e-7*exp(-1*t)+0.1596e-9*exp(-1*t)-0.13261e-7*exp(-1*t),

0.6679e-7*exp(0*t)+0.688e-7*exp(-2*t)+0.2857e-7*exp(-1.999*t)-0.3056e-7*exp(-1.999*t)+0.2328e-9*exp(-1*t)-0.6593e-7*exp(-1*t)-0.67856e-7*exp(-1*t)

```
syms t;
Q=[-0.0012272 0.0001895 0.0009687;0.9998105 -1.0008 0; 0.9990313 0 -
0.9992899];
[V D]=eig(Q);
P0=[1 0 0];
P=P0*V*expm(D*t)*inv(V);
Pt=vpa(P)
```

Pt =

0.9988*exp(-.694e-4*t)+0.506e-3*exp(-1*t)+0.651e-3*exp(-1*t),

0.189e-3*exp(-.694e-4*t)+0.1298e-3*exp(-1*t)-0.319e-3*exp(-1*t),

0.968e-3*exp(-.6936e-4*t)-0.636e-3*exp(-1*t)-0.332e-3*exp(-1*t)

>>

P1=(0.99884288089923034958045305504472/0.00006935770167393707304106381661 4127)+(0.00050612129455726605682202990553425/1.00006091271127517394745609 63572)+(0.00065099780621238538936622010328356/1.00118682958705029761858 89565386)

P1 =

1.4401e+004

<u>The results:</u>

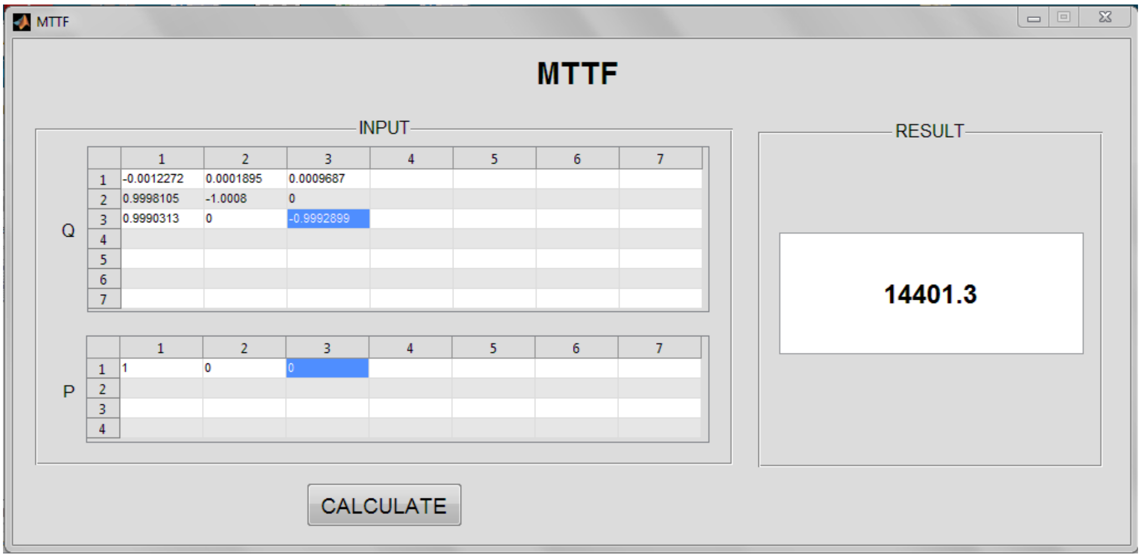

Figure 6-24: GUI for MTTF Calculation

MTTF evaluated by Markov model yields 14401 as compared to theoretical value of 14234.9. This value has been obtained from failure rate of 7.025×10^{-5} per flight hour. This is shown in Figure 6.24. The input is entered on the left hand side of the GUI simulation and can be seen in Figure 6-25. The Figure 6-26 to Figure 6-32 shows the out graph for simulation at different states of the Markov Chain.

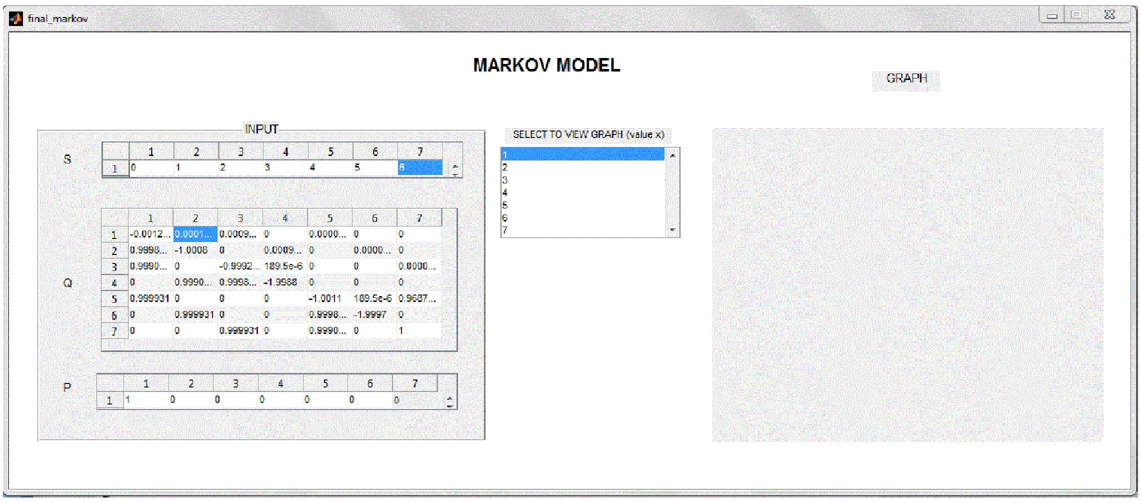

Figure 6-25: Input entered in GUI for Markov Model Simulation

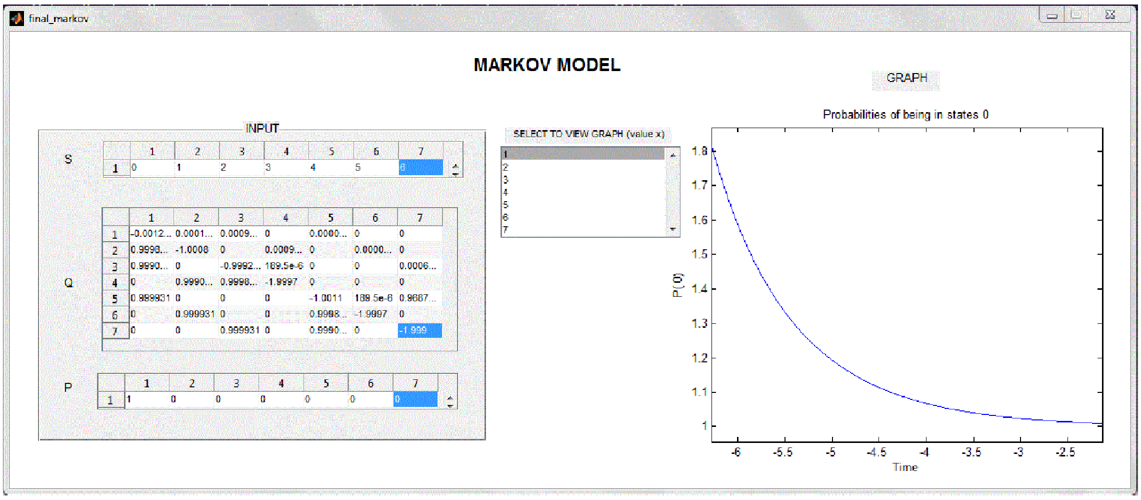

Figure 6-26: Markov Model Simulation at State 1

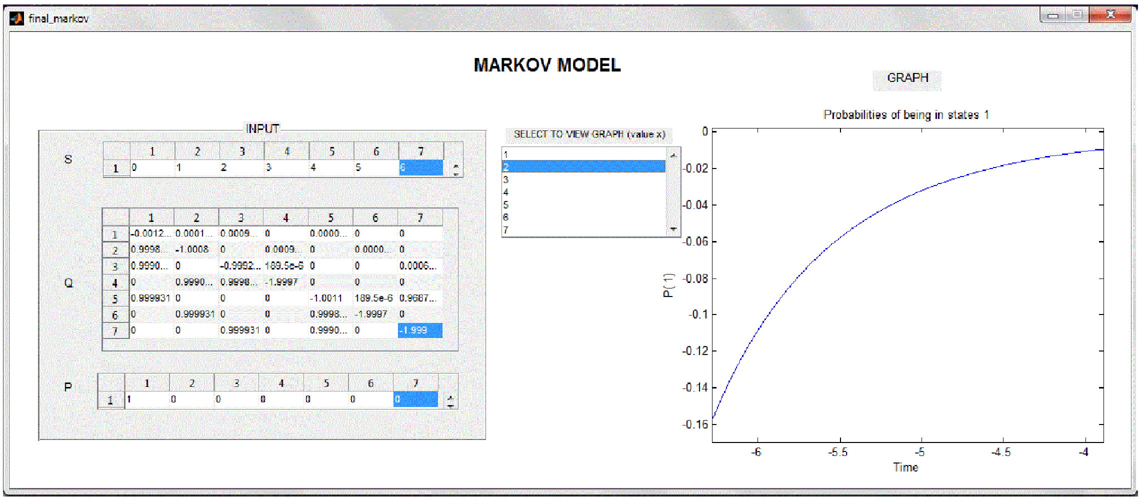

Figure 6-27: Markov Model Simulation at State 2

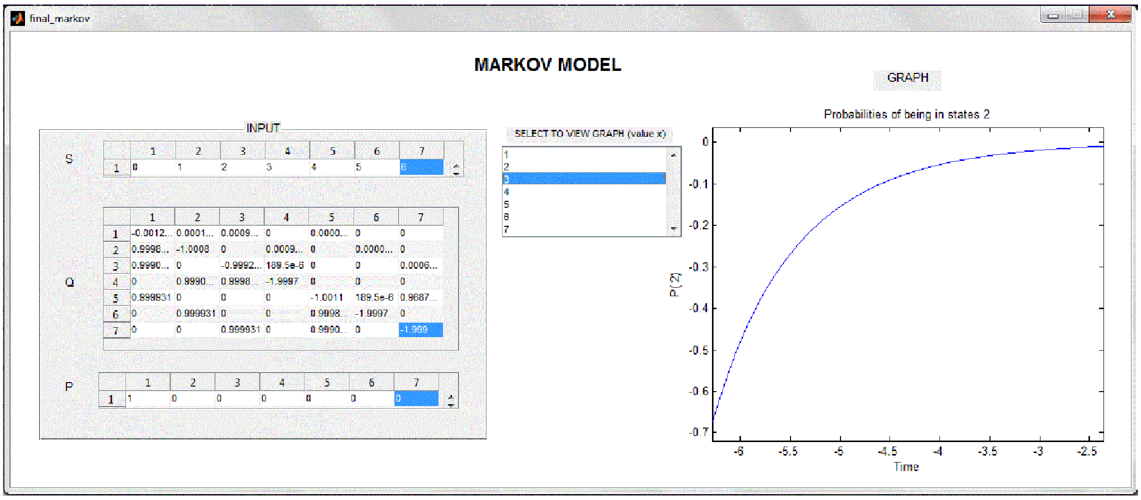

Figure 6-28: Markov Model Simulation at State 3

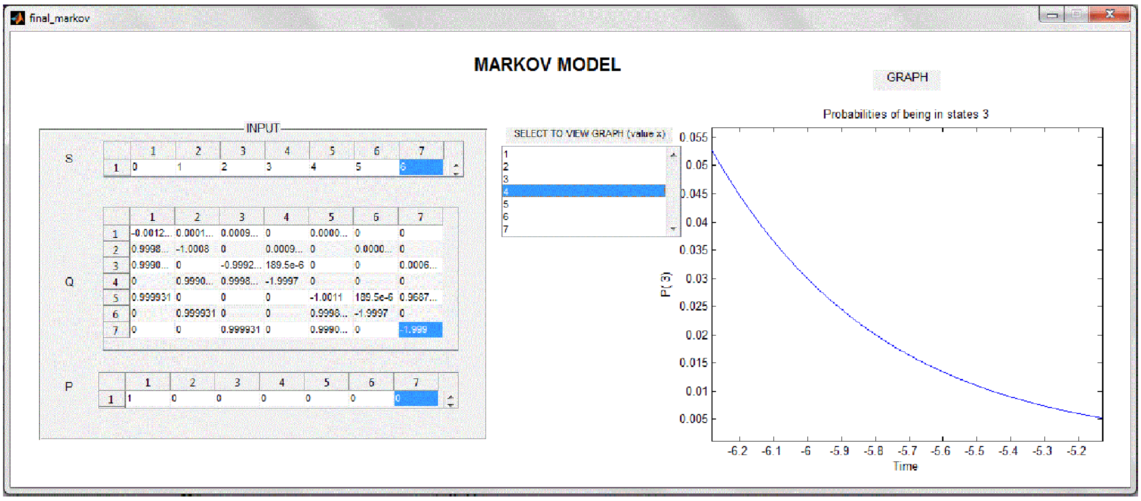

Figure 6-29: Markov Model Simulation at State 4

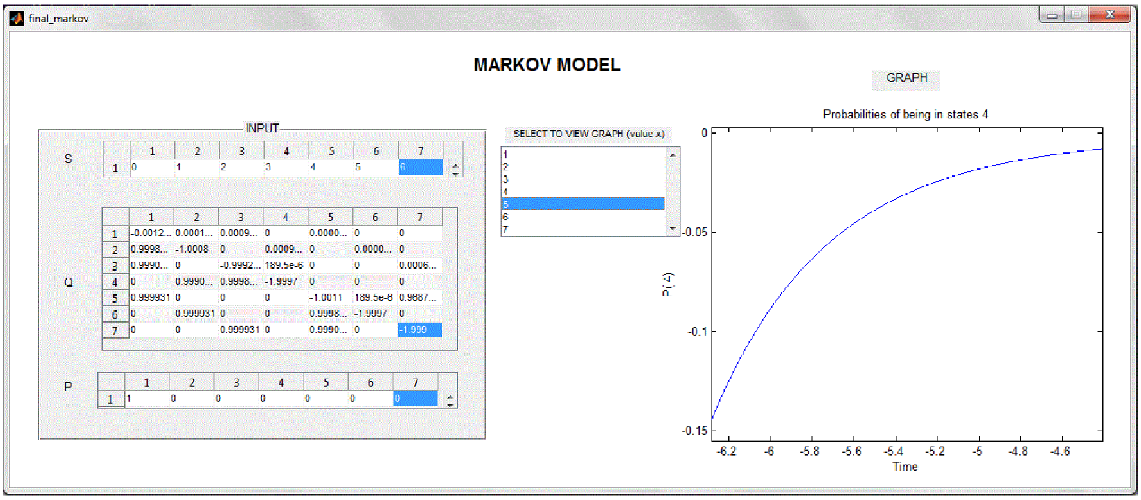

Figure 6-30: Markov Model Simulation at State 5

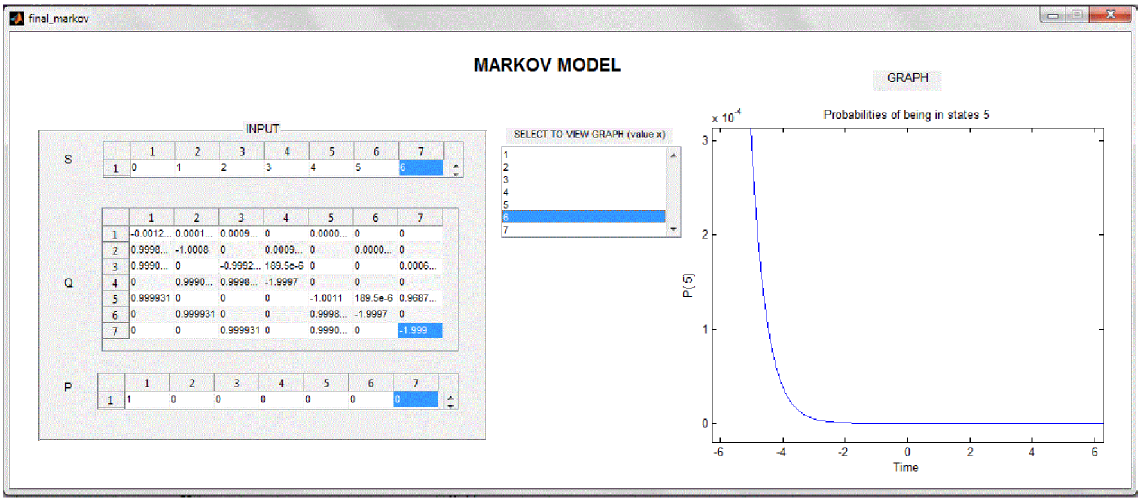

Figure 6-31: Markov Model Simulation at State 6

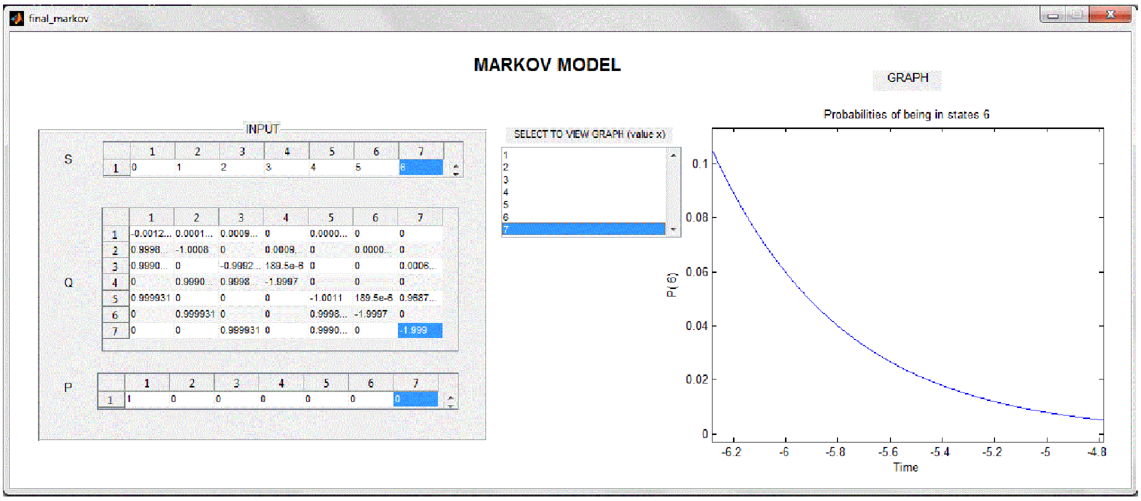

Figure 6-32: Markov Model Simulation at State 7

6.4 Case study using various reliability standards

Table 6-9 describes the different quality for various reliability standards that can be applied to avionics. The highest reliability amongst all four is the MIL-HDBK 217, which most avionics specification refers to. The highest specification in terms of quality has to be for space usage, where faults and failure cannot be tolerated.

Table 6-9: Heritage-based MTBF for small, medium and large characteristic units

(Borer et al., 2010)

Quality	Unit size	MIL-HDBK 217	Relex	RiaC	Vendor DB
Space	Small	183429	183429	183429	183429
	Medium	100000	100000	100000	100000
	Large	52543	52543	52543	52543
Military	Small	18343	50519	146743	141057
	Medium	10000	27542	80000	76900
	Large	5254	14471	42034	40406
Rugged	Small	9171	32741	91715	84561
	Medium	5000	17849	50000	46100
	Large	2627	9379	26272	24222
Communication	Small	4586	14962	36686	42372
	Medium	2500	8157	20000	23100
	Large	1314	4286	10509	12137

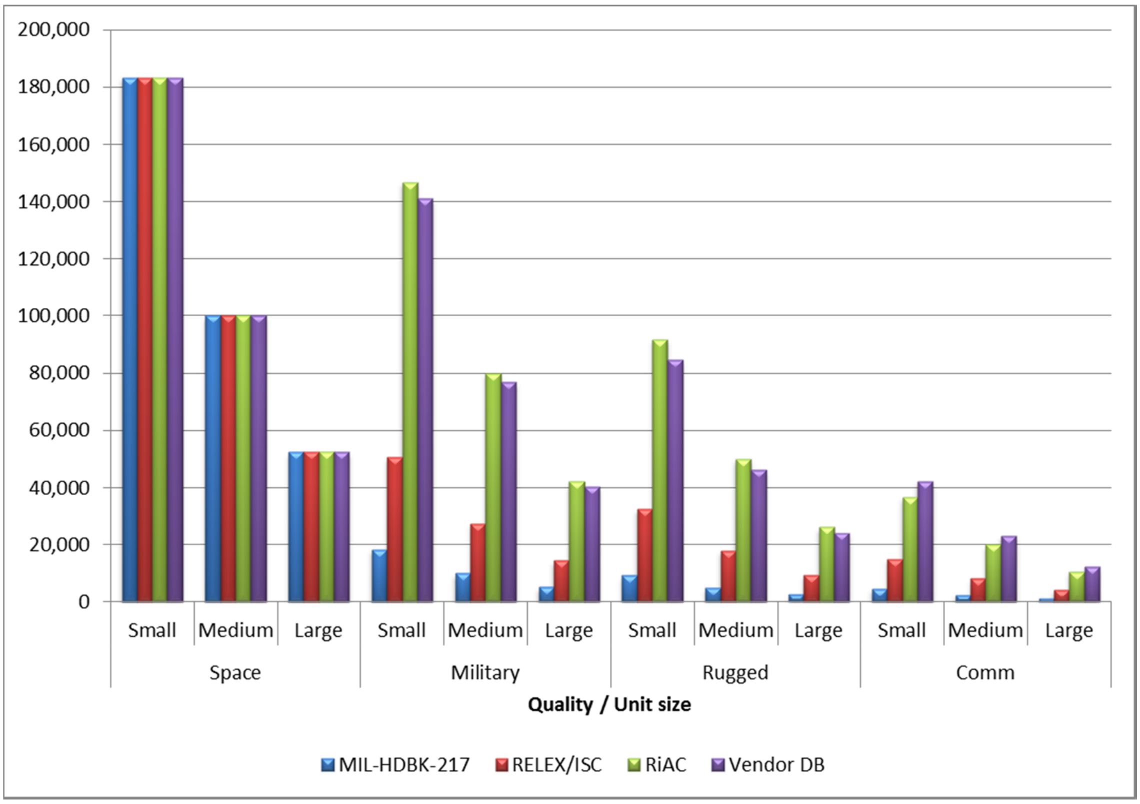

Figure 6-33: Bar graph generated based on the Table 6-9

As shown in Figure 6-33, Space category shows the highest MTBF as compared to the military and rugged category. This reliability number reflects the least expected of failure time of those components or systems.

Table 6-10: Failure rate according to size of EGPWS

	Items	*Failure rate (10^{-6})*
1	Cockpit speakers	2.1
2	Cockpit lamps	4.455
3	Discreet switches	6.37
4	Internal GPS circuit card assembly	10.909
5	TA/Wx Relay	28
6	Global positioning system	85.7
7	Radio altimeter	198.5
8	Data computer	205
9	TA display	227.1
10	Instrument landing system	312

Table 6-10 contains data of EGPWS failure rate which will be cross synthesised with the reliability standards given in Table 6-9. For example, as shown in Table 6-11, a radio altimeter has a failure rate of 198.5(10^{-6}), and according to the reliability standards in Table 6-9, MTBF of 5050.5 falls under the 'large' category.

Table 6-11: Failure rate conversion to MTBF and size

	Failure rate (10^{-6})	MTBF	Component Size
Radio altimeter	198.5	5050.5	large
Data computer	205	4878	large
Instrument landing system	312	3205.1	large
Global positioning system	85.7	11668.6	medium
Internal GPS circuit card assembly	10.909	91667.4	medium
Cockpit lamps	4.455	224466.9	small
Cockpit speakers	2.1	476190.5	small
Discreet switches	6.37	156985.9	small
TA/Wx Relay	28	35714.3	medium
TA display	227.1	4403.3	large

6.4.1 **Failure rate temperature dependent of avionics**

Because all electronics are susceptible to temperature changes, failure rate of avionics will take effect as well. In a study by Vaziry-Zanjany, failure rate increases as junction temperature increases for typical integrated circuit (IC) components.

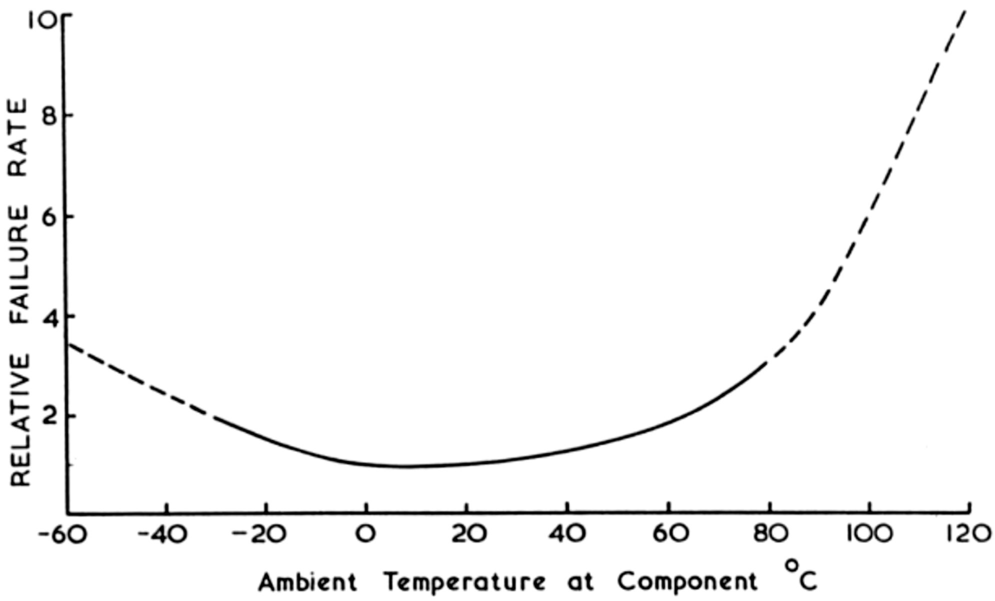

Figure 6-34: Temperature in Celsius versus relative failure rate

The Figure 6-34 describes the average failure rate demonstrated from a typical two extreme groups of avionics components that are highly (graph A) shown in Figure 6-35 and poorly (graph B) as shown in Figure 6-36. According to the study, at 100 degrees Celsius, the rate of failure is $0.4x10^6$ per hour, for component A.

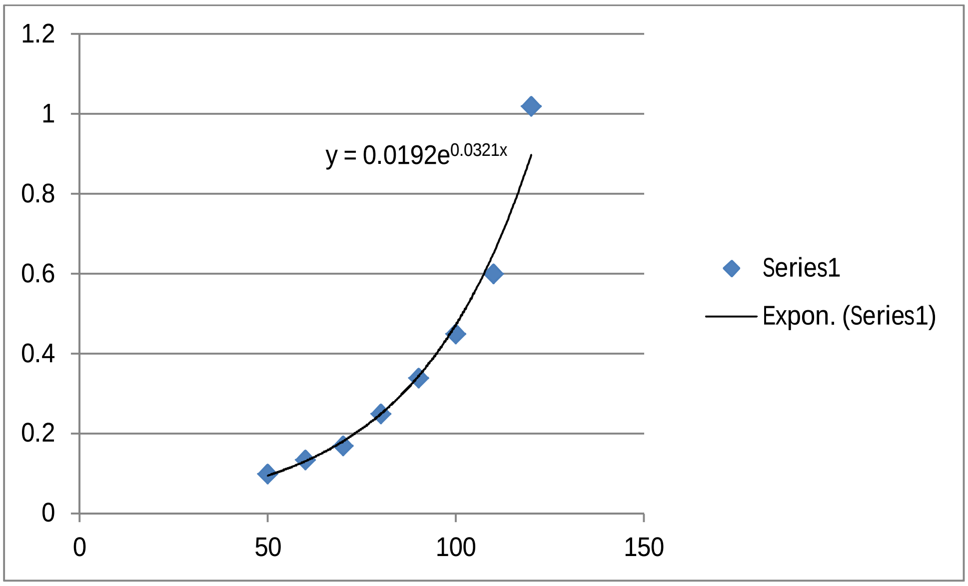

Figure 6-35: Based on Graph A (Vaziry-Zanjany, 1996)

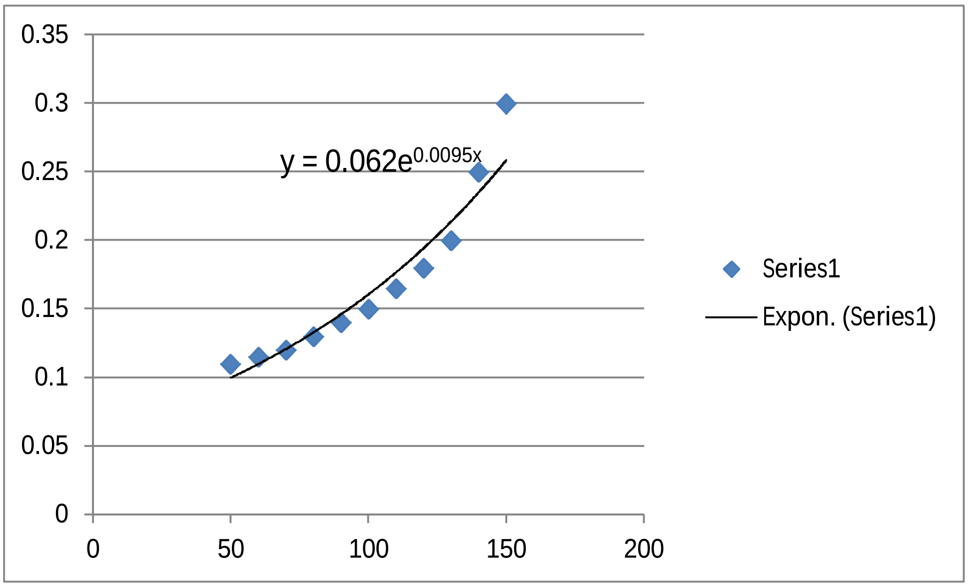

Figure 6-36: Based on graph B (Vaziry-Zanjany, 1996)

Figure 6-37 on the other hand has been averaged based on the two graphs A and B. The failure rate determined to relate failure rate and temperature junction for avionics equipment follows the following formula:

$$\lambda = 0.06058 + 0.00322 \times T_{junction} - 0.00007575 \times T_{junction}{}^{2} + 6.5713\exp(-7) \times T_{junction}{}^{3}$$

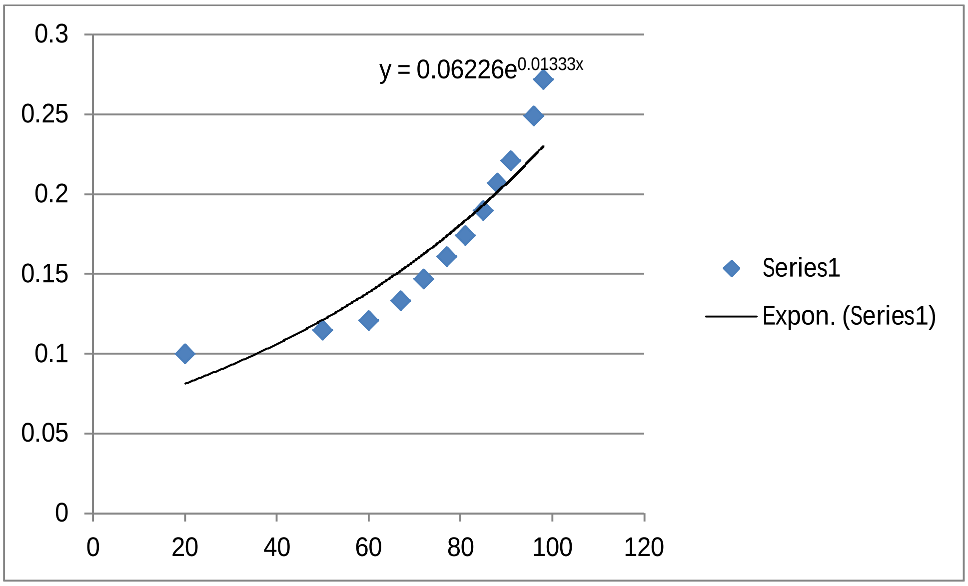

Figure 6-37: Based on Table C-7 (Vaziry-Zanjany, 1996)

In Figure 6-38, the three graphs are shown on the same window.

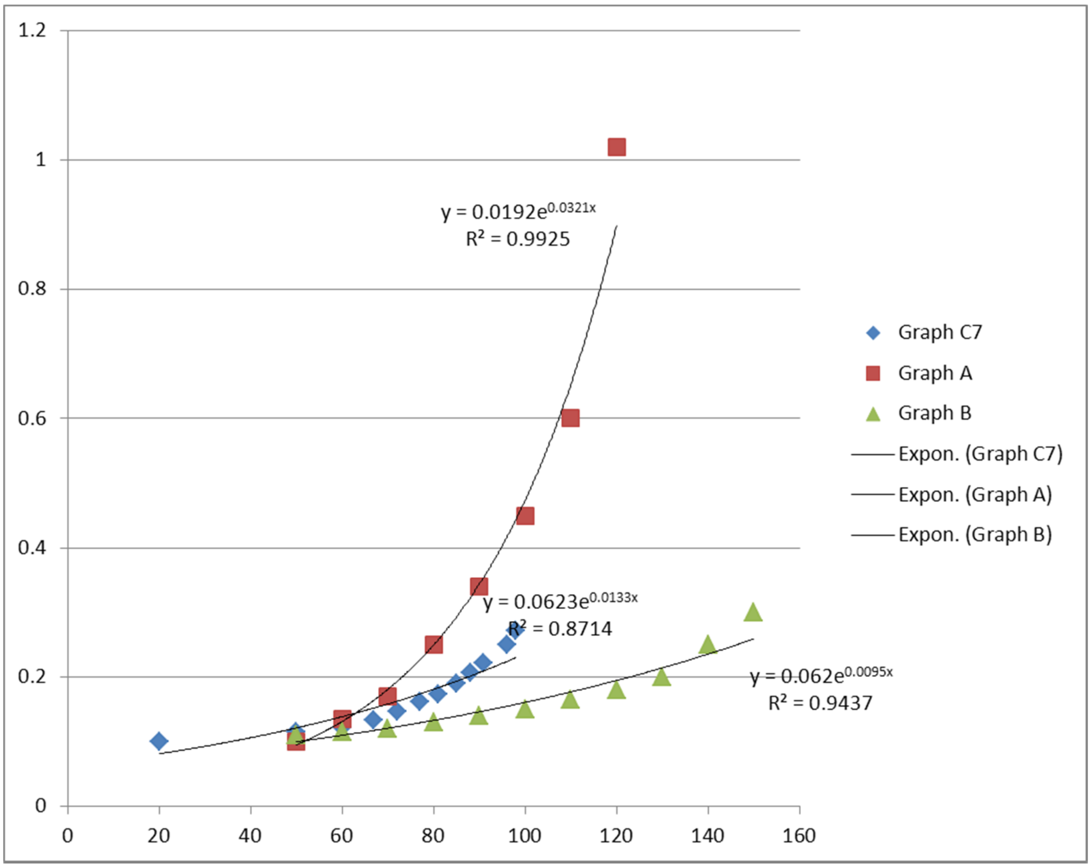

Figure 6-38: The three curves shown on one platform for comparison

The failure rate however, shows very insignificant values where the failure rates were significantly small. With the reliability standards set earlier, failure rates were scaled up by a factor of 10 and simulated again for results. It can be seen in Figure 6-39 that the values were more realistic and can be used to correlate future failure rate-temperature data.

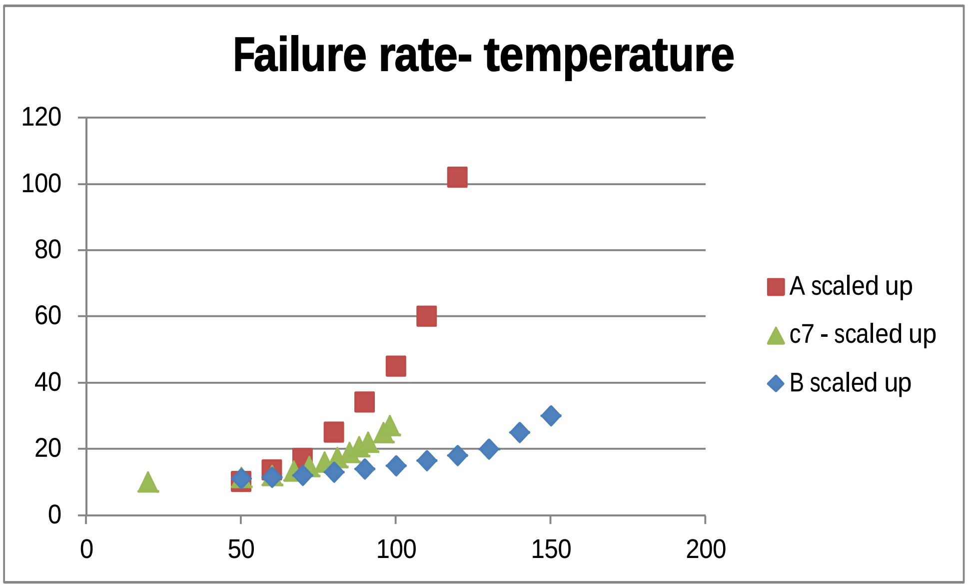

Figure 6-39: The graphs scaled up by a factor of 10

The graphs of scaled up failure rate and temperature were drawn up with the four reliability benchmark of MIL-HDBL, RELEX, RiAC and vendor as shown in Figure 6-40 to Figure 6-43. And the correlation of each graphs were also calculated. From observation RiAC was possibly the closest of the three reliabilty standards where the curve fitted best.

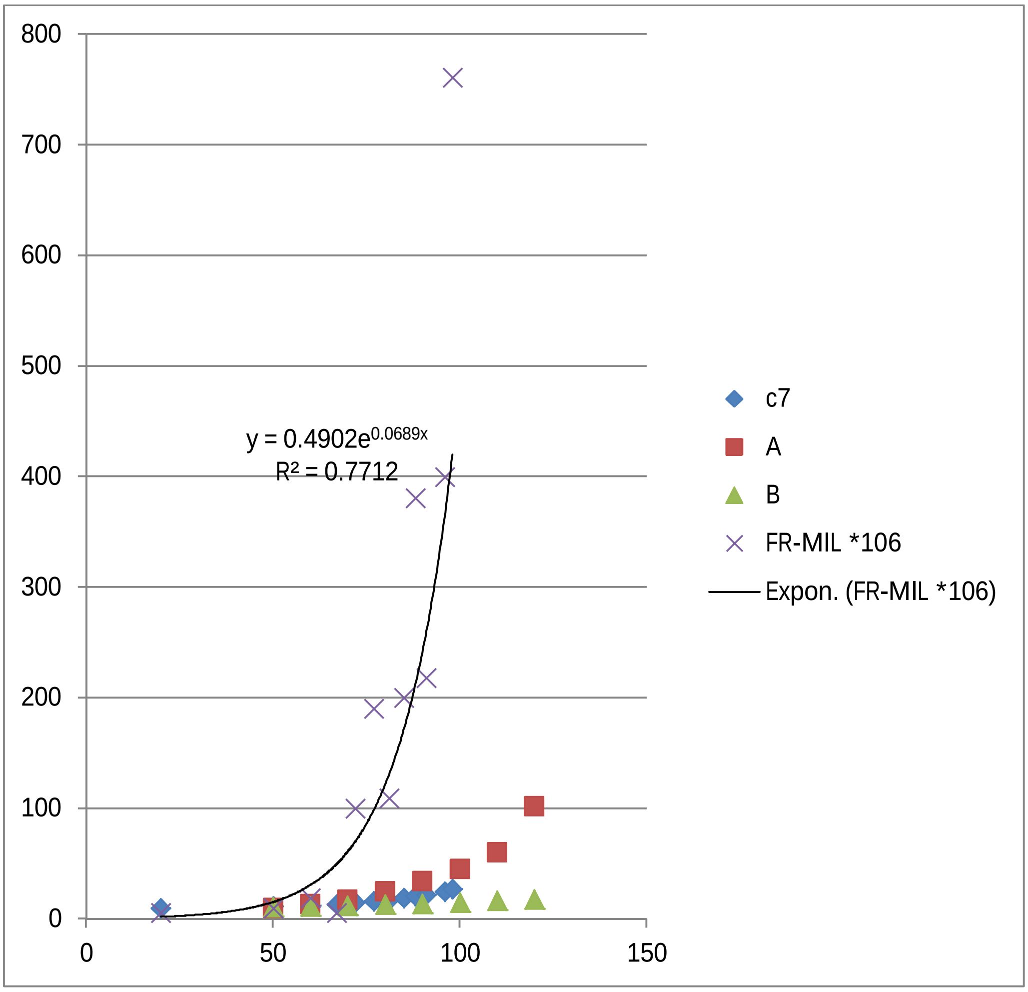

Figure 6-40: The reliability standard of MIL-HDBK with temperature dependent graphs

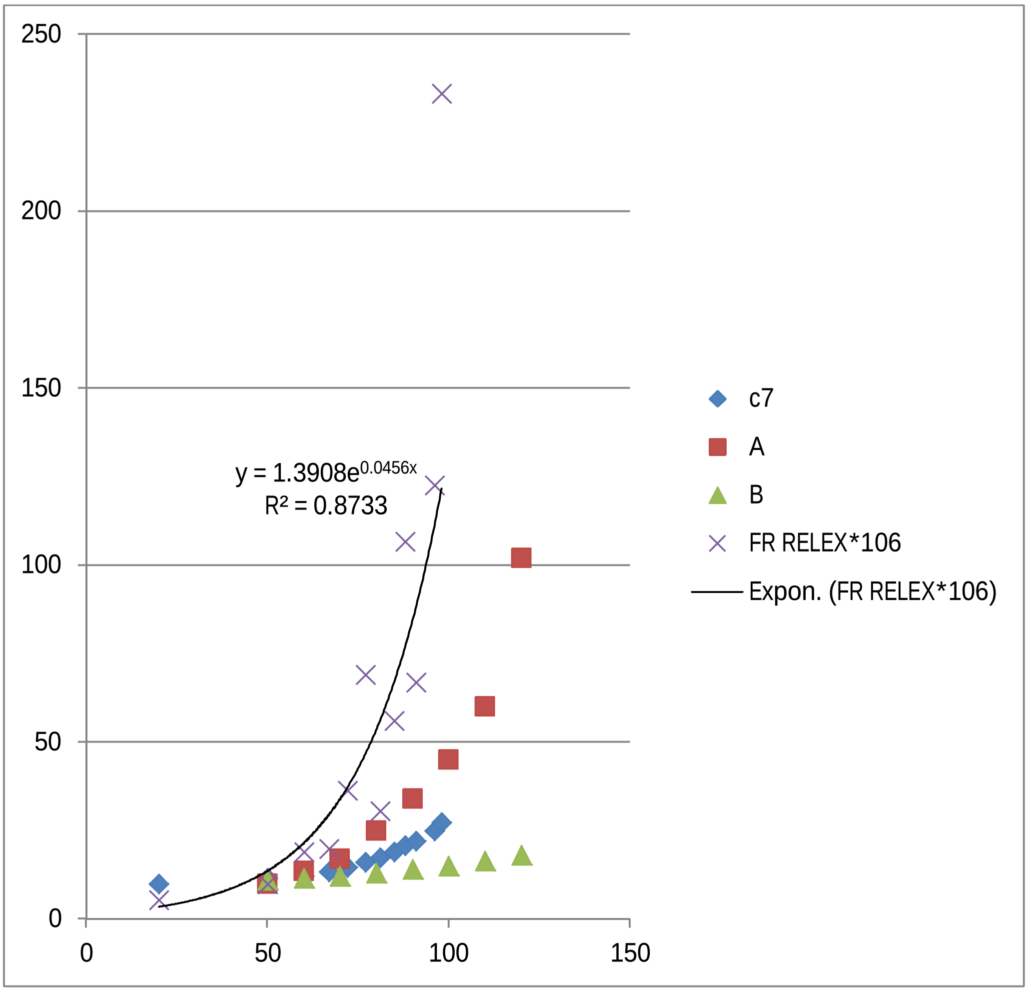

Figure 6-41: The reliability standard of RELEX with temperature dependent graphs

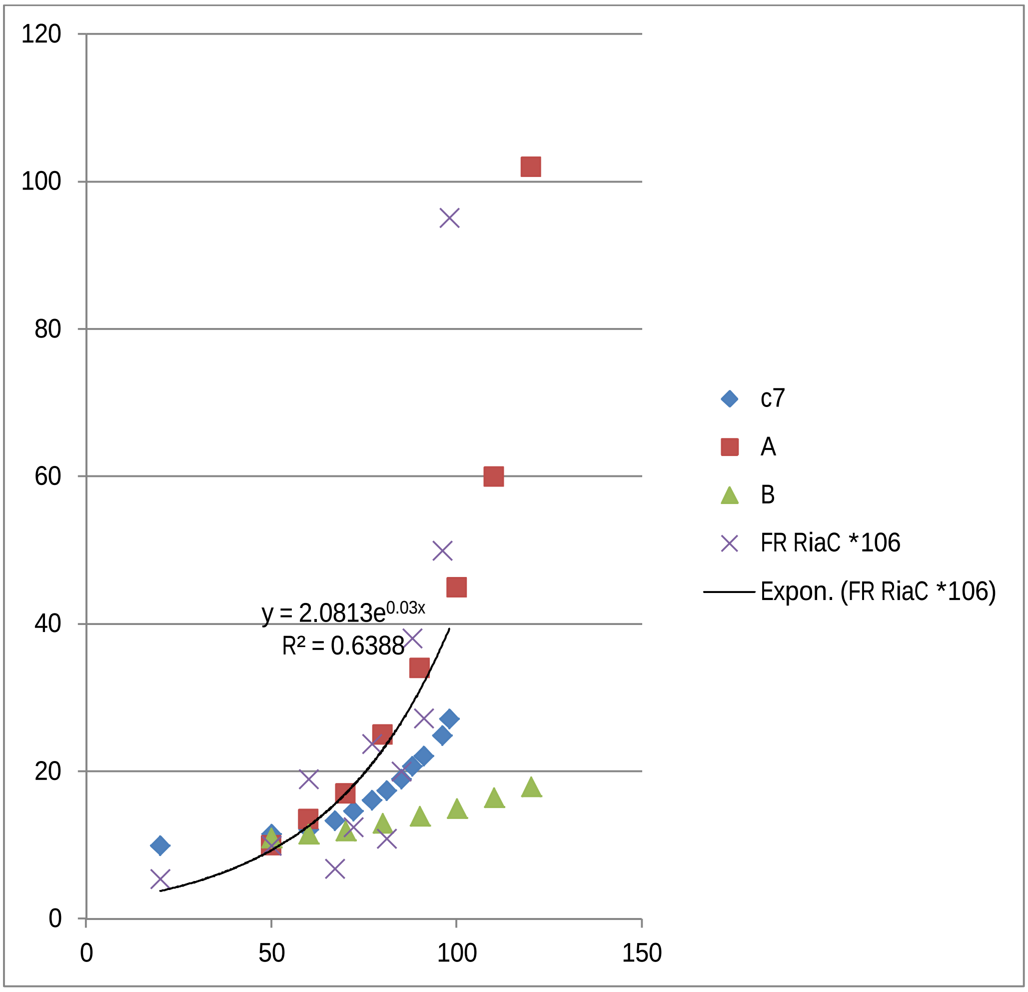

Figure 6-42: The reliability standard of RiAC with temperature dependent graphs

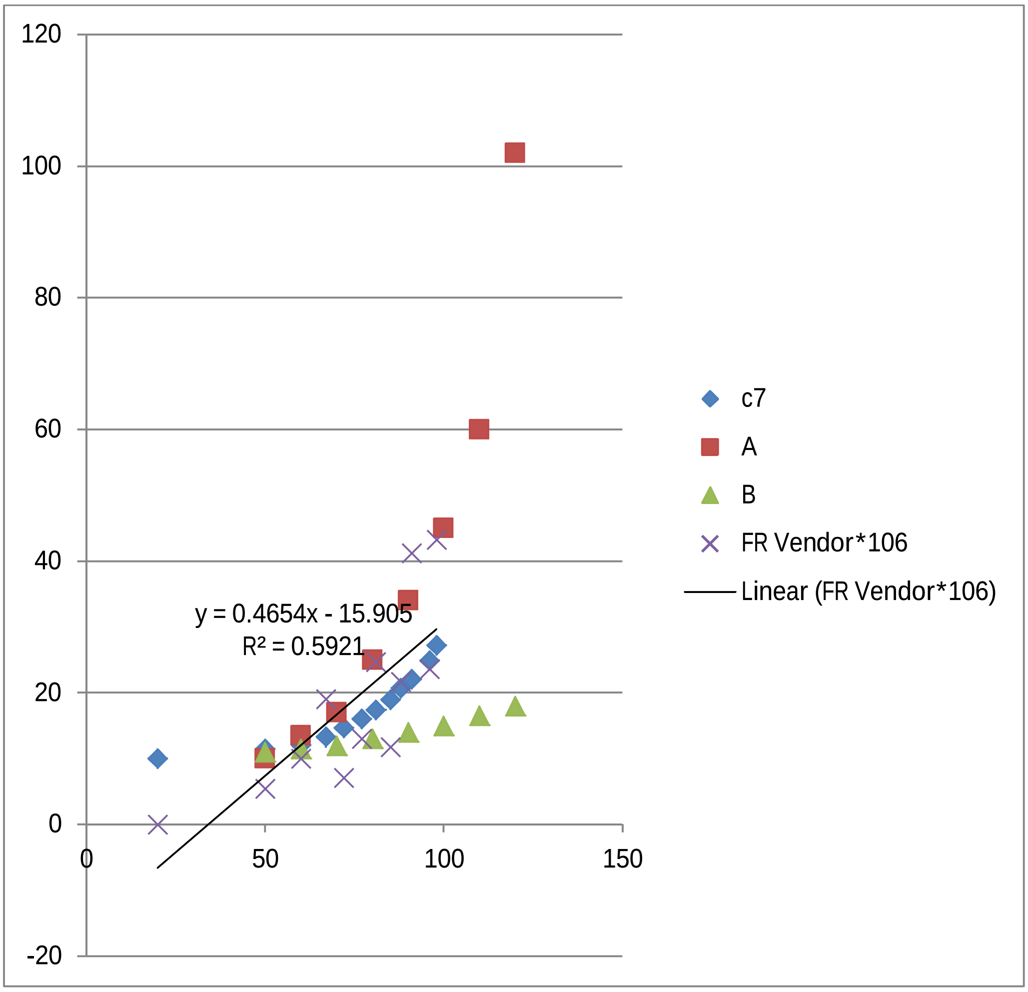

Figure 6-43: The reliability standard of Vendor with temperature dependent graphs

6.4.2 **Correlation analysis**

From the Table 7-12, it can be concluded that the trend or pattern for correlation analysis shows that MIL-HDBK correlates the highest with the EGPWS failure rate data. Similarly, the same result shows in Figure 7-47 where the EGPWS data has the closest pattern to MIL-HDBK as opposed to the other reliability standards shown.

Table 6-12: Correlation analysis using Microsoft Excel for all reliability methods

FR-MIL *106	EGPWS			FR RELEX*10^6	EGPWS			FR RiaC *106	EGPWS			FR Vendor*106	EGPWS	
5.451700658	2.1			5.4517007	2.10			5.451700658	2.1			5.451700658	2.1	
10	4.455			10.0000000	4.46			10	4.455			10	4.455	
19.03203091	6.2			19.0320309	6.20			19.03203091	6.2			19.03203091	6.2	
5.451670937	6.37			19.7945328	6.37			6.81463511	6.37			7.089332681	6.37	
100	9.929			36.3081839	9.93			12.5	9.929			13.00390117	9.929	
190.3311762	28			69.1037247	28.00			23.79026502	28			24.74879968	28	
109.0393632	50			30.5427446	50.00			10.90334187	50			11.82578257	50	
200	68.965			56.0255476	68.97			20	68.965			21.69197397	68.965	
380.6623525	85.7			106.6211750	85.70			38.06333739	85.7			41.28478243	85.7	
218.0549498	87.1			66.8359845	87.10			27.25835469	87.1			23.60049089	87.1	
400	111.6			122.5940910	111.60			50	111.6			43.29004329	111.6	
761.0350076	140			233.3177788	140.00			95.1565325	140			82.39268353	140	
	*FR-MIL *106*	*EGPWS*			*FR RELEX*106*	*EGPWS*			*FR RiaC *106*	*EGPWS*			*FR Vendor*106*	*EGPWS*
FR-MIL *106	1			FR RELEX*106	1			FR RiaC *106	1			FR Vendor*106	1	
EGPWS	0.921674953	1		EGPWS	0.893461245	1		EGPWS	0.863976888	1		EGPWS	0.866825295	1

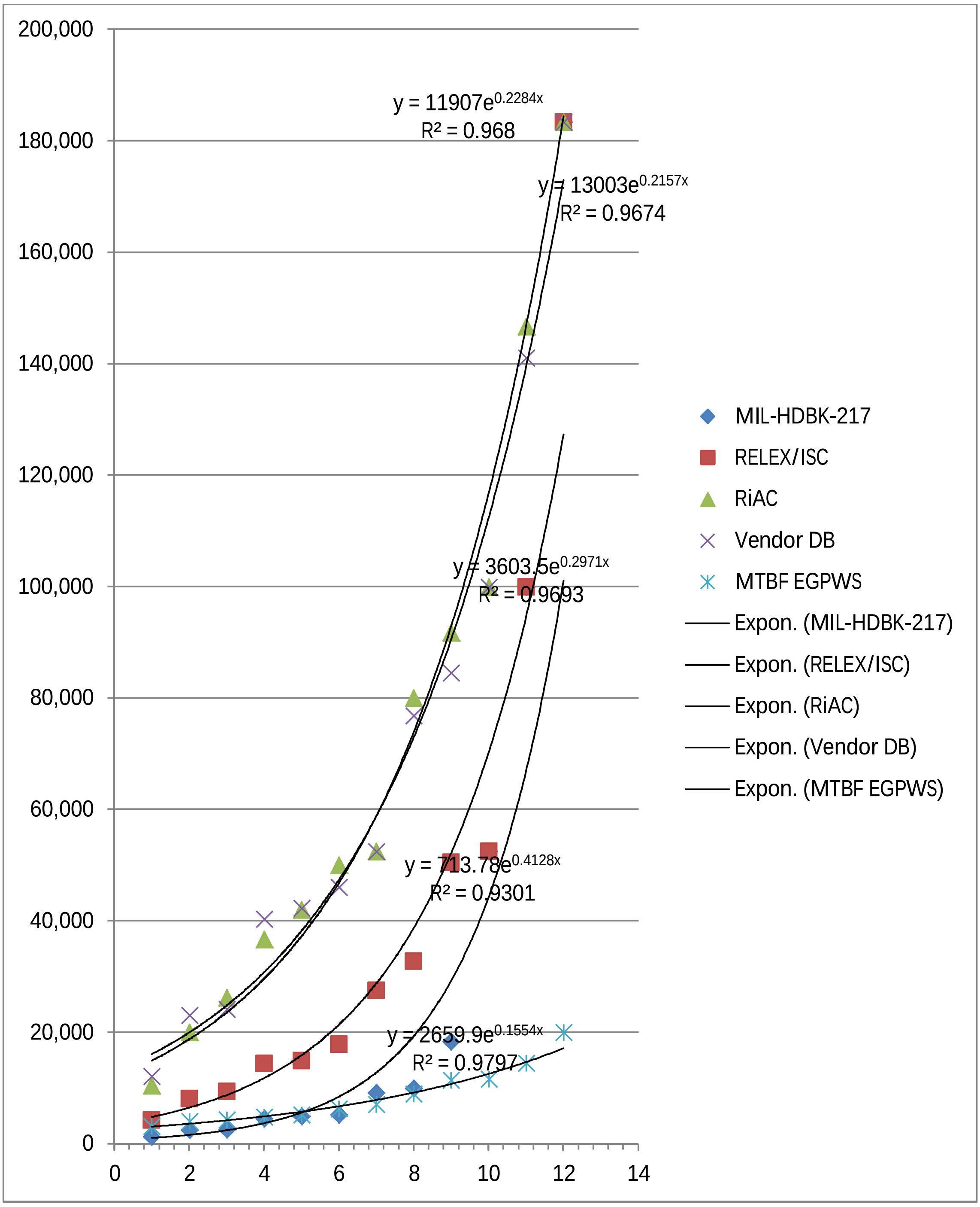

Figure 6-44: MTBF reliability standards with EGPWS components

6.5 SDRS

One method of assessing the removal pattern of an aircraft is by analysing a Service Difficulty Report (SDR). The SDR provides the necessary information.

An SDR is completed for instances of equipment inoperability. SDRs provide information about problems or failures of aircraft components and equipment ("Automated Trend Monitoring for Service Difficulty Reports", 1998). SDRs are completed for each instance of equipment inoperability such as, in-service difficulties, malfunctions, and defects. The data collected and the result from the analysis will be shown in the tables and graphs.

A number of Service Difficulty Report submitted to FAA (Federal Aviation Administration) under Navigation System (ATA 34) installed in Boeing Aircraft. The number of Service Difficulty Report that has been collected is present in the table below categorized according to few set of variables. This research analysed the trend of avionics equipment that is prone to system fault, which includes system malfunction, damaged, unserviceable by using data report from SDR database. Service Difficulty Reports (SDR) consist of maintenance incidents collected by the FAA for the purpose of tracking repair problems with private, commercial and military aircraft and aircraft component. This SDR reports data for the analysis which dated back from 1990 to present. They are largely self-reported by the aircraft owners. The data is reported by tail number and aircraft serial number, so it is possible to trace the maintenance history of the particular airplane with this database.

Table 7.14 shows the number of reports regarding to the number of Service Difficulty Report submitted to FAA (Federal Aviation Administration) under Navigation (ATA 34) installed in Boeing Aircraft. According to the research that has been made, the researcher managed to collect faulty and problems data occurred in the equipment. The reports contain several information includes submitter operation, type of aircraft, problem description, part or structure causing difficulty and etc. Each of the reports submitted to FAA is categorized according to its ATA chapter for ease review.

Table 6-13: List of Service Difficulty Report for Navigation System (ATA 34)

ATA	***Component***	***No. of reports***		***Percentage (%)***
		737 series only	***All Boeing Series***	
3412	Air Data Computer System	96	285	33.68

3417	Altitude Alerting System	1	3	33.33
3428	Inertial Reference System	41	103	39.81
3431	VOR/ILS Navigation System	3	10	30
3441	Weather Radar	300	781	38.41
3442	GPWS	9	60	15
3445	TCAS	6	45	13.33
3448	Radio Altimeter	19	148	12.84
3453	ATC	7	43	16.28
3455	DME	5	40	12.5
3457	ADF	0	14	0
3458	GPS	7	14	50
	Total	**494**	**1546**	**31.95**

6.6 Summary

MTBF and MTBUR are considered key reliability metric or parameter that industries in aerospace and defence use. Even EGPWS manufacturers still refer to MIL HDBK-217F in their manual for EGPWS. MTBF predicts elapse time between what is defined as a failure of system during operation while MTBUR finds the average time (flying hours) that a component functions without the need of any unplanned removal for repair or maintenance. Although it is thought to be appropriate measurement in product reliability it is sometimes even a removal is considered non-trivial as removals can be quite rare.

Table 6-14: Comparison of MTBUR and MTBF values

	MTBUR	MTBF
Product Specification	7000 hours or better	10000 hours or better
Calculation (Probability of loss of all EGPWS functions) top level analysis	N/A	12451.7 hours. (Calculated from known P_{fhr}=8.031x10^{-5} per flight hour. :. λ=8.031x10^{-5} hour
Source 2 (EGPWC)	2202.5 hours	4339.9 hours
Source 2 (TDU)	4949.5 hours	111142 hours
Source 3	2617.3 hours	7046.5 hours

CHAPTER 7

DISCUSSION

This chapter reports the findings by revisiting the presented research aim and objectives. This methodology aims to find the time to failure in order to provide ample time for maintenance personnel to take action before any avionic equipment fail. System level in this context means prognostics will be analysed at the line replaceable unit (LRU) which is a step higher than component level. This research work involves the integration of three research subjects which are prognostics methodology, degradation model and time to failure prediction. In order to achieve the aim of this research, research objectives were identified.

7.1 Achievement of research aim and objectives

The research aims are achieved by:

7.1.1 **Objective 1**

“To analyse the dependency of avionic systems including Line Replaceable Units (LRU) and Line Replaceable Modules (LRM) for fault propagation behaviour degradation”

First of all, it can be concluded that the majority of avionics discrepancies are unscheduled. As a result, flight status will be affected. Secondly, the number of ‘Navigation’ category of removal record was among the highest. The field data on EGPWS recorded an MTBF value which was roughly 3000 hours lower than the published product specification.

7.1.2 **Objective 2**

"To research and develop methods to predict the remaining useful life of avionics LRUs or LRMs"

In the case of simulation of the fault tree diagram transformed into Markov model in achieving the probability of failure in each state of the fault tree, the dependencies among the components in the EGPWS system was established. The failure at each level could be seen to be affecting the probability of failure in relation to other components in the system. With regards to the results, it has been proven that the time to failure estimated by using this method is relatively precise whereby the MTTF evaluated by Markov model yields 14401 hours as compared to theoretical value of 14234.9 hours. This value of 14234.9 hours has been obtained from failure rate of 7.025x10-5 per flight hour.

7.1.3 **Objective 3**

"To research and develop methods to evaluate and predict the degradation performances of avionic systems"

With regards to the reliability standards in avionics, few has been found to be showing strong correlation to the EGPWS MTBF scale of reading. After correlating the EGPWS data sample obtained from the product specification, MIL-HDBK reliability standards shows closest and highest correlation by a factor of 0.922. However, using the failure rate versus temperature standards which has been up scaled by a factor of 10, the EGPWS MTBF showed better correlation with the RIAC and the VENDOR reliability standards.

One of the reasons being, product specification are to be produced with the most rigid quality standards and thus, the EGPWS MTBF value which was extracted from a product specification could have been following the stringent guidelines. The failure rate versus temperature model on the other hand, could have been more realistic although it has been up scaled by a factor of 10, it probably has improved as time passed.

7.1.4 **Objective 4**

“To develop software simulation systems to evaluate methods developed above considering aircraft environment and flight conditions in which avionics experience”.

A matlab graphical user interface software has been created for the methods developed for the prognostics of avionics. This system is named “System Level Airborne Avionic Prognostics”. The two conditions considered were temperature and stress of an EGPWS (airborne avionics) system.

CHAPTER 8

CONCLUSION

This chapter aims to conclude this research by summarising the main contribution, limitations and present the future research in order to fulfil the aim and objectives of this work. It is also in this chapter that the thesis is finalised.

8.1 Main contribution

This research has contributed in investigating on prognostics methodology specifically for airborne avionics system, considering environmental features of temperature and stress as the factors. This study has suggested using different methodologies in finding estimated failure time of avionics equipment in helping MRO overcome logistic issues.

In detail, these are the highlights of this study:

- Critical analysis on different existing approaches suitable for prognostics study. This was done by studying trend of failures through component removal reported by airlines. Most of the reports had shown that majority removals were unscheduled and therefore contributed to major issues such as delays and aircraft on grounds.
- Considered temperature and stress as an environmental factor that affects equipment failure. In this issue, the suggested methodology is by using the Cox's regression analysis though the use of GUI software in seeing the highest possible contributor towards the failure of a system. One other method to predict failure time when field data are available; is the Markov Model.

- Included variety of reliability standards used in common avionics industry. For the study, in relating failure rates and temperature of avionic system, reliability standards such as the ML-HDBK, RELEX, RiAC and vendor were used as a benchmark.

8.1.1 **Limitation to study**

One main limitation of this study is that failure data is commercially sensitive and that limits the availability of data for analysis in this study. Because data is scarce, this study was fitted towards the kinds of data that was available. Another limitation of this study is that avionic equipment deteriorates at many different parts, at different times and into many levels of degradation. So, the definition of the exactly where the prognostics are applied can send different meaning to readers, It is also worth mentioning that this study focuses on system level which has been defined to be the LRU level of avionics. As such, it is hardly possible to actually pin point the exact failure time and as the nearest to it is probably looking into the probability of failure at an instantaneous period of time and probability of failure at different states of time. Other possibilities is to venture in examining the failure modes of equipment to look for common highest incidences and try to apply the methodologies developed on modules.

8.2 Future research

Further research is needed in order to improve on accuracy and precision of estimating failure time. This study should also further be developed for use in system design process, as a built-in rather than an add-on prognostic for avionics. Although real-life case study was used in this research for validation, more information is needed on various avionics LRU so proper justification can be provided to the airlines and OEM. Proposed changes would be feasible if data were more readily available.

8.3 Research conclusion

This study has presented on the approach of prognostics best suited for the airborne avionics systems. Although only top level solution n airborne avionics system has been the focus of this research, it could provide a general view to the maintenance personnel to at least have an idea, when to expect a failure.

The objectives of this research are met and a methodology to predict failure time using field data and product specification were proposed. The research has been unable to provide an intensive result of environmental factors which could have been impressive due to the limitation of sample data and would have required more work and time.

Overall, it can be concluded that this research has enhanced the possibility of improving the avionics maintenance strategy process which plays a major role in the airline industry in general, but specifically focusing on MRO industry.

REFERENCES

Aaseng, G.B., (2001), "Blueprint for an integrated vehicle health management system", *Proceedings of the 20th Conference on Digital Avionics systems*, vol 1, pp.3C1/1-3C1/11.

Abbas, M., Ferri, A.A., Orchard, M.E. and Vachtsevanos, G.J., (2007), "An Intelligent Diagnostic/Prognostic Framework for Automotive Electrical Systems", *Proceedings of IEEE Intelligent Vehicles Symposium.* pp. 352-357

Abraham, Bovas, and Johannes Ledolter. (2009), *Statistical methods for forecasting*. Vol. 234. John Wiley & Sons,

Airbus (2014),
Commercial Aviation Accidents 1958-2013 A Statistical Analysis, Available from: *http://www.airbus.com/company/aircraft-manufacture/quality-and-safety-first/?eID=maglisting_push&tx_maglisting_pi1%5BdocID%5D=39258.* (Accessed: 1st April 2015).

Amit, M., Cavanaugh, K., Pattipati, K., Willett, P. and Galie, T. (2001), "Reasoning and modelling systems in diagnosis and prognosis", *Proceedings of SPIE,* [Online], vol. 4389, no. 194, pp. 15/6/09.

Bird, G., Christensen, M., Lutz, D. and Scandura, P. A., Jr. (2005), "Use of Integrated vehicle health management in the field of commercial aviation", *First International Forum on Integrated System Health Engineering and Managementin Aerospace, 2005,* [Online], no. Paper #12.

Borer, N., Claypool, I., Clark, D., West, J., Odegard, R., Somervill, K. and Suzuki, N. (2010), "Model-driven development of reliable avionics architectures for Lunar Surface Systems", *Aerospace Conference, 2010 IEEE,* IEEE, pp. 1.

Byington, C.S., Kalgren, P.W., Dunkin, B.K. and Donovan, B.P., (2004), "Advanced diagnostic/prognostic reasoning and evidence transformation techniques for improved avionics maintenance", *Proceedings of IEEE Aerospace Conference*, vol 5, pp. 3434

Czerwonka, Stephen Paul. (2000), "Avionics life-cycle forecasting model", Massachusetts Institute of Technology Doctor of Philosophy Thesis.

Dale, C.J. (1985), "Application of the proportional hazards model in the reliability field", *Reliability Engineering,* vol. 10, no. 1, pp. 1-14.

Coble, J. B. (2010), "Merging Data Sources to Predict Remaining Useful Life–An Automated Method to Identify Prognostic Parameters", Cranfield University Masters of Science Thesis.

Coit, D. W., Evans, J. L., Vogt, N. T. and Thompson, J. R. (2005), "A Method for Correlating Field Life Degradation with Reliability Prediction for Electronic Modules", *Quality and Reliability Engineering International,* vol. 21, no. 7, pp. 715-726.

Denson, W. (1998), "The history of reliability prediction", *Reliability, IEEE Transactions on,* vol. 47, no. 3, pp. SP321-SP328.

Eurocontrol (2009),
Delays to Air Transport in Europe, Digest – April 2009.

Fang Tu, Ghoshal, S., Jianhui Luo, Biswas, G., Mahadevan', S., Jaw, L. and Navarra, K., (2007), "PHM Integration with Maintenance and Inventory Management Systems", *Proceedings of IEEE Aerospace Conference*, pp. 1-12

Ghobbar, A. A. and Friend, C. H. (2002), "Sources of intermittent demand for aircraft spare parts within airline operations", *Journal of Air Transport Management,* vol. 8, no. 4, pp. 221-231.

Harun, M. K. (1998), *Reliability analysis for cost-effective Maintenance with application to Aircraft Engine Components* (unpublished Masters of Science thesis), Glasgow Caledonian University, Glasgow, Scotland.

Hecht, H. (2006), "Prognostics for electronic equipment: an economic perspective", *Reliability and Maintainability Symposium, 2006. RAMS '06. Annual,* pp. 165.

Hines, J. W. and Usynin, A. (2008), "Current Computational Trends in Equipment Prognostics", *International Journal of Computational Intelligence,* vol. 1, no. 1, pp. 94.

Janasak, K. M. and Beshears, R. R. (2007), "Diagnostics To Prognostics - A Product Availability Technology Evolution", *Reliability and Maintainability Symposium, 2007. RAMS '07. Annual,* pp. 113.

Jaw, L. C. and Wang, W. (2006), "Mathematical Formulation of model-based methods for diagnostics and prognostics", *ASME Turbo Expo 2006: Power for Land, Sea and Air,* May 8-11, 2006, Barcelona, Spain, ASME.

Jianhui Luo, Namburu, M., Pattipati, K., Liu Qiao, Kawamoto, M. and Chigusa, S., (2003), "Model-based prognostic techniques [maintenance applications]", *Proceedings of IEEE Systems Readiness Technology AUTOTESCON*, pp. 330-340

Jie Gu, Vichare, N., Tracy, T. and Pecht, M., (2007), "Prognostics Implementation Methods for Electronics", *Reliability and Maintainability Symposium*, pp. 101-106

Johnson, D. M. (1996), "A review of fault management techniques used in safety-critical avionic systems", *Progress in Aerospace Sciences,* vol. 32, no. 5, pp. 415-431.

Justice, A. C., Covinsky, K. E. and Berlin, J. A. (1999), "Assessing the generalizability of prognostic information", *Annals of Internal Medicine,* vol. 130, no. 6, pp. 515-524.

Kalgren, Byington, C. Byington, and B. Donovan. (2004), "Automated Data Translation for Avionics Health Management through Intelligent Evidence Streams and Databases." In *AHS Annual Forum*.

Kumar, U. Dinesh, John Crocker, Jezdimir Knezevic, and Mohamed El-Haram. (2012) *Reliability, Maintenance and Logistic Support:-A Life Cycle Approach*. Springer Science & Business Media.

Lou, Z., Li, H., He, Y. and Du, J. (2009), "Fault Diagnosis and Prediction for Airborne Electronic Equipment ", *Failure Analysis and Prevention,* vol. 1.

Moir, I., Seabridge, A. and Jukes, M. (2013), *Civil avionics systems,* John Wiley & Sons.

Ofsthun, S. (2002), "Integrated vehicle health management for aerospace platforms", *Instrumentation & Measurement Magazine, IEEE,* vol. 5, no. 3, pp. 21-24.

Orchard, M.E. and Vachtsevanos, G.J., (2007), *A particle filtering-based framework for real-time fault diagnosis and failure prognosis in a turbine engine*.

Pecht, M. G. (2008), *Prognostics and Health Management of Electronics,* 1st ed, John Wiley & Sons, New Jersey.

Pipe, K., (2008), "Practical prognostics for Condition Based Maintenance", *Proceedings of Prognostics and Health Management International Conference*. pp. 1-10

Roemer, M., Byington, C. S., Kacprzynski, G. J. and Vachtsevanos, G. (2006), "An overview of selected Prognostic technologies with application to engine health management", *American Society of Mechanical Engineers,* [Online], .

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S. and Schwabacher, M., (2008), "Metrics for evaluating performance of prognostic techniques", *Proceedings of Prognostics and Health Management International Conference*. pp. 1-17

Schwabacher, M. A. (2005), "A Survey of Data-Driven Prognostics", *Infotech@Aerospoace; Arlington, VA; USA; 26-29 Sept. 2005,* Vol. AIAA Paper 2005-7002, American Institute of Aeronautics and Astronautics, 1801 Alexander Bell Drive, Suite 500, Reston, VA, 20191-4344, USA, [URL:http://www.aiaa.org], .

Sheppard, J.W., Kaufman, M.A. and Wilmering, T.J., (2008), "IEEE standards for prognostics and health management". *Proceedings of IEEE Systems Readiness Technology AUTOTESCON*, pp. 97-103

Shyur, H. (2008), "A quantitative model for aviation safety risk assessment", *Computers & Industrial Engineering,* vol. 54, no. 1, pp. 34-44.

Sundström, T., Mesgarzadeh, B., Krysander, M., Klein, M., Söderquist, I., Crona, A., Fransson, T. and Alvandpour, A. (2008), "Prognostics of electronic systems through power supply current trends", *IEEE Internatioanl Conference on Prognostics and Health Management,* Denver, USA, .

Tuchband, B. A. (2007), "Implementation of prognostics and health management for electronic systems", University of Maryland, College Park Master of Science Thesis.

Uckun, S., Goebel, K. and Lucas, P.J.F., (2008), "Standardizing research methods for prognostics", *International Conference on Prognostics and Health Management*, pp. 1-10.

Vachtsevanos, G. (2006), *Intelligent fault diagnosis and prognosis for engineering systems,* Wiley, Hoboken, N.J.

Vaziry-Zanjany, M. A. F. (1996), "Aircraft conceptual design modelling incorporating reliability and maintainability predictions", Cranfield University Doctor of Philosophy Thesis.

White, M. (2008), "Microelectronics reliability: physics-of-failure based modeling and lifetime evaluation", *Jet Propulsion Laboratory, California Institute of Technology, Pasadena, California,* JPL Publication 08-5 2/08.

Wilkinson, C., Humphrey, D., Vermeire, B., and Houston, J. (2004), "Prognostic and health management for avionics", *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, vol. 5, pp. 3435-3447

Wu, H., Liu, Y., Ding, Y. and Liu, J. (2004), "Methods to reduce direct maintenance costs for commercial aircraft", *Aircraft Engineering Aerospace Technology,* vol. 76, no. 1, pp. 15.

APPENDIX 1

GUI For Kaplan Meier Chart Source Code

```
function varargout = kaplan_meier(varargin)
% KAPLAN_MEIER MATLAB code for kaplan_meier.fig
%      KAPLAN_MEIER, by itself, creates a new KAPLAN_MEIER or raises
the existing
%      singleton*.
%
%      H = KAPLAN_MEIER returns the handle to a new KAPLAN_MEIER or
the handle to
%      the existing singleton*.
%
%      KAPLAN_MEIER('CALLBACK',hObject,eventData,handles,...) calls
the local
%      function named CALLBACK in KAPLAN_MEIER.M with the given input
arguments.
%
%      KAPLAN_MEIER('Property','Value',...) creates a new KAPLAN_MEIER
or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before kaplan_meier_OpeningFcn gets called.
An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to kaplan_meier_OpeningFcn via
varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help kaplan_meier

% Last Modified by GUIDE v2.5 27-Jan-2016 17:09:46

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @kaplan_meier_OpeningFcn, ...
                   'gui_OutputFcn',  @kaplan_meier_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
```

```
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before kaplan_meier is made visible.
function kaplan_meier_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to kaplan_meier (see VARARGIN)

% Choose default command line output for kaplan_meier
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes kaplan_meier wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = kaplan_meier_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
%ST
dataQ=get(handles.uitable1, 'data')
Q = str2double(dataQ)
%detect empty rows
empty_rowsQ = all( isnan(Q), 2 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_rowsQ == false, 1, 'last')
empty_rowsQ(1:last_nonempty) = false
%remove them
Q(empty_rowsQ,:) = []
empty_columnQ = all( isnan(Q), 1 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_columnQ == false, 1, 'last')
empty_columnQ(1:last_nonempty) = false
%remove them
```

```
Q(:,empty_columnQ) = []
ST= Q'

%number at risk
dataX=get(handles.uitable2, 'data')
X = str2double(dataX)
%detect empty rows
empty_rowsX = all( isnan(X), 2 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_rowsX == false, 1, 'last')
empty_rowsX(1:last_nonempty) = false
%remove them
X(empty_rowsX,:) = []
empty_columnX = all( isnan(X), 1 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_columnX == false, 1, 'last')
empty_columnX(1:last_nonempty) = false
%remove them
X(:,empty_columnX) = []
NumberAtRisk= X'

%number of failure
dataP=get(handles.uitable3, 'data')
P = str2double(dataP)
%detect empty rows
empty_rowsP = all( isnan(P), 2 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_rowsP == false, 1, 'last')
empty_rowsP(1:last_nonempty) = false
%remove them
P(empty_rowsP,:) = []
empty_columnP = all( isnan(P), 1 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_columnP == false, 1, 'last')
empty_columnP(1:last_nonempty) = false
%remove them
P(:,empty_columnP) = []
NumberofFailure= P'

E=1-(NumberofFailure./NumberAtRisk)
cdf=zeros(1,10)';
cdf(1)=E(1) %initialize cdf(1)=E

axes(handles.axes1);
for i=2:9;
    cdf(i)=E(i)*cdf(i-1)
    i=i+1
end
stairs(ST,cdf)
```

APPENDIX 2

GUI For Simple Markov Source Code

```
function varargout = markov_simple_final(varargin)
% MARKOV_SIMPLE_FINAL MATLAB code for markov_simple_final.fig
%      MARKOV_SIMPLE_FINAL, by itself, creates a new
MARKOV_SIMPLE_FINAL or raises the existing
%      singleton*.
%
%      H = MARKOV_SIMPLE_FINAL returns the handle to a new
MARKOV_SIMPLE_FINAL or the handle to
%      the existing singleton*.
%
%      MARKOV_SIMPLE_FINAL('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in MARKOV_SIMPLE_FINAL.M with the given
input arguments.
%
%      MARKOV_SIMPLE_FINAL('Property','Value',...) creates a new
MARKOV_SIMPLE_FINAL or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before markov_simple_final_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to markov_simple_final_OpeningFcn
via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
markov_simple_final

% Last Modified by GUIDE v2.5 25-Jan-2016 20:36:09

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @markov_simple_final_OpeningFcn,
...
                   'gui_OutputFcn',  @markov_simple_final_OutputFcn,
...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
```

```
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before markov_simple_final is made visible.
function markov_simple_final_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to markov_simple_final (see
VARARGIN)

% Choose default command line output for markov_simple_final
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes markov_simple_final wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = markov_simple_final_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
dataX=get(handles.uitable1, 'data')
X = str2double(dataX)
%detect empty rows
empty_rowsX = all( isnan(X), 2 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_rowsX == false, 1, 'last')
empty_rowsX(1:last_nonempty) = false
%remove them
X(empty_rowsX,:) = []
```

```
empty_columnX = all( isnan(X), 1 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_columnX == false, 1, 'last')
empty_columnX(1:last_nonempty) = false
%remove them
X(:,empty_columnX) = []

dataP=get(handles.uitable2, 'data')
P = str2double(dataP)
%detect empty rows
empty_rowsP = all( isnan(P), 2 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_rowsP == false, 1, 'last')
empty_rowsP(1:last_nonempty) = false
%remove them
P(empty_rowsP,:) = []

empty_columnP = all( isnan(P), 1 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_columnP == false, 1, 'last')
empty_columnP(1:last_nonempty) = false
%remove them
P(:,empty_columnP) = []

Xsteady=[P'-
eye(size(P));ones(1,length(P))]\[zeros(length(P),1);1]%Probability of
state at maximum limit
After2steps= X*P
After3steps= After2steps*P

set(handles.uitable3, 'data', Xsteady);
set(handles.uitable4, 'data', After2steps);
set(handles.uitable5, 'data', After3steps);
```

APPENDIX 3

GUI For Markov with Output Graph Source Code

```
function varargout = MARKOV_WITH_GRAPH(varargin)
% MARKOV_WITH_GRAPH MATLAB code for MARKOV_WITH_GRAPH.fig
%      MARKOV_WITH_GRAPH, by itself, creates a new MARKOV_WITH_GRAPH
or raises the existing
%      singleton*.
%
%      H = MARKOV_WITH_GRAPH returns the handle to a new
MARKOV_WITH_GRAPH or the handle to
%      the existing singleton*.
%
%      MARKOV_WITH_GRAPH('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in MARKOV_WITH_GRAPH.M with the given
input arguments.
%
%      MARKOV_WITH_GRAPH('Property','Value',...) creates a new
MARKOV_WITH_GRAPH or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before MARKOV_WITH_GRAPH_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to MARKOV_WITH_GRAPH_OpeningFcn
via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help MARKOV_WITH_GRAPH

% Last Modified by GUIDE v2.5 27-Jan-2016 22:22:02

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @MARKOV_WITH_GRAPH_OpeningFcn,
...
                   'gui_OutputFcn',  @MARKOV_WITH_GRAPH_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
```

```
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before MARKOV_WITH_GRAPH is made visible.
function MARKOV_WITH_GRAPH_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to MARKOV_WITH_GRAPH (see
VARARGIN)

% Choose default command line output for MARKOV_WITH_GRAPH
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes MARKOV_WITH_GRAPH wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = MARKOV_WITH_GRAPH_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on selection change in listbox1.
function listbox1_Callback(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns listbox1
contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
listbox1

dataS=get(handles.uitable1, 'data')
S = str2double(dataS)
%detect empty rows
empty_rowsS = all( isnan(S), 2 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_rowsS == false, 1, 'last')
```

```
empty_rowsS(1:last_nonempty) = false
%remove them
S(empty_rowsS,:) = []
empty_columnS = all( isnan(S), 1 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_columnS == false, 1, 'last')
empty_columnS(1:last_nonempty) = false
%remove them
S(:,empty_columnS) = []

dataQ=get(handles.uitable2, 'data')
Q = str2double(dataQ)
%detect empty rows
empty_rowsQ = all( isnan(Q), 2 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_rowsQ == false, 1, 'last')
empty_rowsQ(1:last_nonempty) = false
%remove them
Q(empty_rowsQ,:) = []
empty_columnQ = all( isnan(Q), 1 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_columnQ == false, 1, 'last')
empty_columnQ(1:last_nonempty) = false
%remove them
Q(:,empty_columnQ) = []

[V D]=eig(Q)
syms t;

dataP0=get(handles.uitable3, 'data')
P0 = str2double(dataP0)
%detect empty rows
empty_rowsP0 = all( isnan(P0), 2 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_rowsP0 == false, 1, 'last')
empty_rowsP0(1:last_nonempty) = false
%remove them
P0(empty_rowsP0,:) = []
empty_columnP0 = all( isnan(P0), 1 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_columnP0 == false, 1, 'last')
empty_columnP0(1:last_nonempty) = false
%remove them
P0(:,empty_columnP0) = []

P=P0*V*expm(D*t)*inv(V);
Pt=vpa(P)
N=length(S)

a=get(handles.listbox1, 'Value');
if(a==1)
    axes(handles.axes1);
    x=1
    Pt(x)
    ezplot(Pt(x));
    ylabel(['P( ' num2str(S(x)) ')']);
    xlabel('Time')
    title('Probabilities of being in states 0');
```

```
elseif(a==2)
    axes(handles.axes1);
    x=2
    Pt(x)
    ezplot(Pt(x));
    ylabel(['P( ' num2str(S(x)) ')']);
    xlabel('Time')
    title('Probabilities of being in states 1');
elseif(a==3)
    axes(handles.axes1);
    x=3
    Pt(x)
    ezplot(Pt(x));
    ylabel(['P( ' num2str(S(x)) ')']);
    xlabel('Time')
    title('Probabilities of being in states 2')
elseif(a==4)
    axes(handles.axes1);
    x=4
    Pt(x)
    ezplot(Pt(x));
    ylabel(['P( ' num2str(S(x)) ')']);
    xlabel('Time')
    title('Probabilities of being in states 3');
elseif(a==5)
    axes(handles.axes1);
    x=5
    Pt(x)
    ezplot(Pt(x));
    ylabel(['P( ' num2str(S(x)) ')']);
    xlabel('Time')
    title('Probabilities of being in states 4');
elseif(a==6)
    axes(handles.axes1);
    x=6
    Pt(x)
    ezplot(Pt(x));
    ylabel(['P( ' num2str(S(x)) ')']);
    xlabel('Time')
    title('Probabilities of being in states 5');
elseif(a==7)
    axes(handles.axes1);
    x=7
    Pt(x)
    ezplot(Pt(x));
    ylabel(['P( ' num2str(S(x)) ')']);
    xlabel('Time')
    title('Probabilities of being in states 6');
end

% --- Executes during object creation, after setting all properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: listbox controls usually have a white background on Windows.
```

```
%        See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

APPENDIX 4

GUI For Mean Time to Failure Source Code

```
function varargout = MTTF(varargin)
% MTTF MATLAB code for MTTF.fig
%      MTTF, by itself, creates a new MTTF or raises the existing
%      singleton*.
%
%      H = MTTF returns the handle to a new MTTF or the handle to
%      the existing singleton*.
%
%      MTTF('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in MTTF.M with the given input
arguments.
%
%      MTTF('Property','Value',...) creates a new MTTF or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before MTTF_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to MTTF_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help MTTF

% Last Modified by GUIDE v2.5 27-Jan-2016 17:39:12

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @MTTF_OpeningFcn, ...
                   'gui_OutputFcn',  @MTTF_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT
```

```matlab
% --- Executes just before MTTF is made visible.
function MTTF_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to MTTF (see VARARGIN)

% Choose default command line output for MTTF
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes MTTF wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = MTTF_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

dataQ=get(handles.uitable1, 'data')
Q = str2double(dataQ)
%detect empty rows
empty_rowsQ = all( isnan(Q), 2 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_rowsQ == false, 1, 'last')
empty_rowsQ(1:last_nonempty) = false
%remove them
Q(empty_rowsQ,:) = []
empty_columnQ = all( isnan(Q), 1 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_columnQ == false, 1, 'last')
empty_columnQ(1:last_nonempty) = false
%remove them
Q(:,empty_columnQ) = []

dataP0=get(handles.uitable2, 'data')
P0 = str2double(dataP0)
%detect empty rows
empty_rowsP0 = all( isnan(P0), 2 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_rowsP0 == false, 1, 'last')
```

```
empty_rowsP0(1:last_nonempty) = false
%remove them
P0(empty_rowsP0,:) = []
empty_columnP0 = all( isnan(P0), 1 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_columnP0 == false, 1, 'last')
empty_columnP0(1:last_nonempty) = false
%remove them
P0(:,empty_columnP0) = []

syms t;
[V D]=eig(Q);
P=P0*V*expm(D*t)*inv(V);
Pt=vpa(P)
F = int(Pt(1),t,0,Inf)
mttf=single(F)
set(handles.result,'String',mttf)

function result_Callback(hObject, eventdata, handles)
% hObject    handle to result (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of result as text
%        str2double(get(hObject,'String')) returns contents of result
as a double


% --- Executes during object creation, after setting all properties.
function result_CreateFcn(hObject, eventdata, handles)
% hObject    handle to result (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

APPENDIX 5

GUI for Constant Pressure Source Code

```
function varargout = CONSTANT_PRESSURE(varargin)
% CONSTANT_PRESSURE MATLAB code for CONSTANT_PRESSURE.fig
%      CONSTANT_PRESSURE, by itself, creates a new CONSTANT_PRESSURE
or raises the existing
%      singleton*.
%
%      H = CONSTANT_PRESSURE returns the handle to a new
CONSTANT_PRESSURE or the handle to
%      the existing singleton*.
%
%      CONSTANT_PRESSURE('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in CONSTANT_PRESSURE.M with the given
input arguments.
%
%      CONSTANT_PRESSURE('Property','Value',...) creates a new
CONSTANT_PRESSURE or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before CONSTANT_PRESSURE_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to CONSTANT_PRESSURE_OpeningFcn
via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help CONSTANT_PRESSURE

% Last Modified by GUIDE v2.5 27-Jan-2016 23:36:12

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @CONSTANT_PRESSURE_OpeningFcn,
...
                   'gui_OutputFcn',  @CONSTANT_PRESSURE_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
```

```
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before CONSTANT_PRESSURE is made visible.
function CONSTANT_PRESSURE_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to CONSTANT_PRESSURE (see
VARARGIN)

% Choose default command line output for CONSTANT_PRESSURE
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes CONSTANT_PRESSURE wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = CONSTANT_PRESSURE_OutputFcn(hObject, eventdata,
handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
dataQ=get(handles.uitable1, 'data')
Q = str2double(dataQ)
%detect empty rows
empty_rowsQ = all( isnan(Q), 2 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_rowsQ == false, 1, 'last')
empty_rowsQ(1:last_nonempty) = false
%remove them
Q(empty_rowsQ,:) = []
empty_columnQ = all( isnan(Q), 1 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_columnQ == false, 1, 'last')
empty_columnQ(1:last_nonempty) = false
```

```
%remove them
Q(:,empty_columnQ) = []
%select column
xdatatemp = Q(:,[1])
xdatatemp2 = Q(:,[2])

[b, log1, H, stats]=coxphfit(xdatatemp2,xdatatemp)
hazardRatio=char(vpa(exp(b),4))
set(handles.edit1,'String',num2str(stats.covb,4))
set(handles.edit2,'String',num2str(stats.beta,4))
set(handles.edit3,'String',num2str(stats.se,4))
set(handles.edit4,'String',num2str(stats.z,4))
set(handles.edit5,'String',num2str(stats.p,4))
set(handles.edit6,'String',hazardRatio)


function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1
as a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2
as a double


% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3
as a double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6
as a double


% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
```

```
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6
as a double


% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

APPENDIX 6

GUI for Constant Temperature Source Code

```
function varargout = CONSTANT_TEMPERATURE(varargin)
% CONSTANT_TEMPERATURE MATLAB code for CONSTANT_TEMPERATURE.fig
%      CONSTANT_TEMPERATURE, by itself, creates a new
CONSTANT_TEMPERATURE or raises the existing
%      singleton*.
%
%      H = CONSTANT_TEMPERATURE returns the handle to a new
CONSTANT_TEMPERATURE or the handle to
%      the existing singleton*.
%
%      CONSTANT_TEMPERATURE('CALLBACK',hObject,eventData,handles,...)
calls the local
%      function named CALLBACK in CONSTANT_TEMPERATURE.M with the
given input arguments.
%
%      CONSTANT_TEMPERATURE('Property','Value',...) creates a new
CONSTANT_TEMPERATURE or raises the
%      existing singleton*.  Starting from the left, property value
pairs are
%      applied to the GUI before CONSTANT_TEMPERATURE_OpeningFcn gets
called.  An
%      unrecognized property name or invalid value makes property
application
%      stop.  All inputs are passed to CONSTANT_TEMPERATURE_OpeningFcn
via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
CONSTANT_TEMPERATURE

% Last Modified by GUIDE v2.5 27-Jan-2016 23:31:17

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @CONSTANT_TEMPERATURE_OpeningFcn,
...
                   'gui_OutputFcn',  @CONSTANT_TEMPERATURE_OutputFcn,
...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end
```

```
if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before CONSTANT_TEMPERATURE is made visible.
function CONSTANT_TEMPERATURE_OpeningFcn(hObject, eventdata, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to CONSTANT_TEMPERATURE (see
VARARGIN)

% Choose default command line output for CONSTANT_TEMPERATURE
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes CONSTANT_TEMPERATURE wait for user response (see
UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = CONSTANT_TEMPERATURE_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;


% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
dataQ=get(handles.uitable1, 'data')
Q = str2double(dataQ)
%detect empty rows
empty_rowsQ = all( isnan(Q), 2 )
%but we only want to strip trailing empty rows
last_nonempty = find(empty_rowsQ == false, 1, 'last')
empty_rowsQ(1:last_nonempty) = false
%remove them
Q(empty_rowsQ,:) = []
empty_columnQ = all( isnan(Q), 1 )
```

```
%but we only want to strip trailing empty rows
last_nonempty = find(empty_columnQ == false, 1, 'last')
empty_columnQ(1:last_nonempty) = false
%remove them
Q(:,empty_columnQ) = []
%select column
xdatatemp = Q(:,[1])
xdatatemp2 = Q(:,[2])

[b, log1, H, stats]=coxphfit(xdatatemp2,xdatatemp)
hazardRatio=char(vpa(exp(b),4))
set(handles.edit1,'String',num2str(stats.covb,4))
set(handles.edit2,'String',num2str(stats.beta,4))
set(handles.edit3,'String',num2str(stats.se,4))
set(handles.edit4,'String',num2str(stats.z,4))
set(handles.edit5,'String',num2str(stats.p,4))
set(handles.edit6,'String',hazardRatio)


function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1
as a double


% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2
as a double


% --- Executes during object creation, after setting all properties.
```

```
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%        str2double(get(hObject,'String')) returns contents of edit3
as a double


% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%        str2double(get(hObject,'String')) returns contents of edit4
as a double


% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end



function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%        str2double(get(hObject,'String')) returns contents of edit6
as a double


% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```