POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

ALGEBRAIC DYNAMIC FAULT TREE ANALYSIS FOR AVIONIC SYSTEMS

NEDA BAGHALIZADEH MOGHADAM
Département de génie électrique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie électrique

Août 2019

© Neda Baghalizadeh Moghadam, 2019.

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Ce mémoire intitulé :

ALGEBRAIC DYNAMIC FAULT TREE ANALYSIS FOR AVIONIC SYSTEMS

présenté par **Neda BAGHALIZADEH MOGHADAM**
en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Guchuan ZHU, président
Yves AUDET, membre et directeur de recherche
Yvon SAVARIA, membre et codirecteur de recherche
François LEDUC-PRIMEAU, membre

DEDICATION

I would like to dedicate this thesis to my family especially to my dear son Arshavin.

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincere gratitude to my supervisors Prof. Yves Audet, and Prof. Yvon Savaria for their continuous support of my master's program and related research, for their motivation, patience, and immense knowledge. Their wise guidance enlightened me the first glance of research and helped me throughout my studies and writing of this thesis. I could not have imagined having better advisers and mentors for my program.

Besides my supervisors, my sincere thanks also go to Madame Nathalie Levesque, Madam Marie-Yanick Laplante and Mr. Jean Bouchard who preciously assisted me with administrative and technical tasks during my program.

I sincerely acknowledge MITACS Scholarship Council, for offering me a partial scholarship to continue my studies in Canada.

Finally, I would like to thank my parents, my spouse and my little son Arshavin for supporting me spiritually throughout my master's program and my life continuously..

RÉSUMÉ

Les analyses par arbres de défaillance se basent sur une représentation graphique des événements probabilistes qui peuvent engendrer la panne d'un système quelconque. Dans l'analyse traditionnelle, les événements sont considérés comme étant statistiquement indépendants et sont reliés entre eux par la logique combinatoire. L'arbre de défaillance permet de déterminer les relations existantes entre les différents événements et leurs différents modes de fonctionnement, de façon à déduire analytiquement une probabilité de panne du système. A chaque événement est associé un taux de panne.

Les portes représentant les relations de la logique combinatoire : AND, OR,...etc, sont incapables de représenter convenablement la relation entre différents événements dont la séquence d'apparition est importante. Nous avons alors affaire à un arbre de défaillance dit "dynamique" qui ne peut être modélisé sans le recours de portes logiques dynamiques. L'arbre de défaillance dynamique est alors modélisé à partir de portes dynamiques et de portes logiques standards ou statiques. Parmi les portes dynamiques les plus utilisées, on retrouve la porte priorité-AND (PAND), la porte séquentielle (SEQ), la porte de rechange ou en attente (SPARE) et la porte de dépendance fonctionnelle (FDEP). Ces portes permettent de conceptualiser un arbre de défaillance entre des événements ayant une certaine causalité.

Au cours des dernières décennies, plusieurs accidents d'avion ont été rapportés dont la cause était reliée au système de pilotage. La modélisation du système de pilotage par un arbre de défaillance dynamique est un bon moyen d'évaluer la probabilité de panne et d'améliorer la robustesse du système.

Ce mémoire porte sur la conception d'un arbre de défaillance dynamique pour un système avionique simple. Une méthode algébrique est utilisée pour calculer la probabilité de panne du système global. Des théorèmes mathématiques sur les portes dynamiques sont utilisés pour développer un modèle Matlab qui calcule les différentes probabilités de panne des sous-systèmes et la probabilité globale. Comparé aux méthodes traditionnelles de résolution des arbres de défaillance dynamique, soient les méthodes Monte-Carlo et les chaines de Markov, la méthode algébrique proposée dans ce travail est beaucoup plus simple à implémenter et demande moins de temps de calcul, tout en conservant une précision semblable sur les résultats.

ABSTRACT

Fault Tree Analysis (FTA) is a method mainly based on a graphical representation of different combinations of basic events which result in a Top Event (TE). In the traditional analysis, the TE occurrences are logically investigated by considering all possible events statistically independent implementing logical Boolean gates. A fault tree is used to determine the operational relationship among different components under different modes to derive an analytical expression giving the probability of failure. Since traditional logical gates, such as AND, OR, etc., are not capable to demonstrate the dynamic behavior of failure mechanisms in a system with dependent events and failures, a Dynamic FTA (DFTA) is used as a promising method. A DFTA employs dynamic gates in its fault tree to represent a sequential notation for the system failures by considering the order of occurrence and different combinations of basic events. The dynamic relationships between basic events can be analyzed using DFT, whereas the traditional static FT expresses only the probability of failure as a function of basic event failure rates. A DFT has at least one dynamic gate along with a combination of static gates such as AND ($\cdot$), OR ($+$), and voting gates. The most commonly used dynamic gates in DFTA are; the priority AND (PAND) gate, the sequence enforcing (SEQ) gate, the standby or spare (Spare) gate, and the functional dependency (FDEP) gate. Dynamic gates in fault trees provide a conceptually simple modeling framework to represent system-level Failure in terms of interactions between component failures through basic events relationships.

Some fatal risks in flight control are typically due to unforeseen failures in different sections of the systems, particularly, in flight control hardware and software systems. Modeling them necessitates a reliable quantitative analysis such as FTA that dynamically predicts faults in the related components. The use of dynamic gates enables the analysis to be more dependable and accurate compared to conventional FTA. This thesis investigates the design of a reliable DFT for an avionic system by taking into account the least reliable flight components. An algebraic method for solving a DFT with a promising algorithm is demonstrated to determine the failure probability of the aircraft which is referred to as the top event (TE) based on the relationships between many basic events. To this end, using temporal operators and some related theorems, the behavioral model of the dynamic gates and the subtrees in the DFT are determined. Using the obtained behavioral models, the probabilistic models of the dynamic gates and subtrees in the DFT are determined. Finally, the total probability of failure is calculated through algebraic analysis of the subtrees in the structure and their individual failure probabilities. Compared to traditional methods, such as Monte Carlo simulations

and Markov analysis, our proposed algebraic method is simpler to implement with lower computational work load and less computer intensive to solve a DFT rapidly and accurately.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

FT	Fault tree
FTA	Fault tree analysis
DFTA	Dynamic fault tree analysis
TE	Top event
PAND	Priority AND
SEQ	Sequence enforcing
FDEP	Functional dependency
pdf	Probability density function
cdf	Cumulative distribution function

CHAPTER 1 INTRODUCTION

Despite remarkable scientific and technological advances in the avionic systems of aircrafts, failures in different subsystems of the aircrafts, such as faults in hardware or software, are still one of the major causes of fatal flight accidents worldwide. Defects in control system of an aircraft may lead to irrecoverable flight accidents. In this regard, Fault Tree Analysis (FTA) has gained a great deal of attention as a method to model failure mechanisms in many applications such as control structures of an aircraft, where quantitative reliability and safety analysis are crucial [2–4]. A Fault Tree (FT) exploits logical and probabilistic analysis for all realistic ways through which an undesired faulty state called Top Event (TE) can happen in a given system. When a system possesses several types of dynamic metrics, the corresponding FT becomes larger, which results in a more complex analysis. Depending on the required accuracy, the structural function of the system can be defined in diverse ways. A more efficient and reliable method is to use a combination of dynamic gates along with static ones to construct a Dynamic Fault Tree (DFT) [5]. In Dynamic Fault Tree Analysis (DFTA), not only combination of failure events but also their order of occurrence is considered. Considering the order of failures in such a Boolean model allows for analyzing the dynamic relationships between the TE and the basic events, which cannot be carried out by traditional static FT that only expresses the occurrence of basic events. It should be stressed that solving a fault tree requires precise algorithms to obtain the failure probability of the TE based on basic events.

A conventional FTA is a hierarchical, extensive failure analysis in which the occurrence of the TE in a system is analyzed by using Boolean logic gates to model a series of lower level (Basic) events and their relationships. Such a structure is used to model the system failure probability based on that of individual subtrees. Using FTA, the potential causes of failures of a given system are identified and the probability of the TE is evaluated. Faults can originate from several sources, such as component hardware, human operations, software, etc. Such an analysis is represented by a graphical structure called a FT that describes the logical interrelationships between the basic events which can cause the undesired TE. The fault tree is represented by a TE, several basic events and the gates that connect the events to each other. The top gate describes the system failure as a TE resulting from the lower basic events in a tree-shape network. Basic events cause basic failures in the system. The top event is analyzed throughout the fault tree using its constituent basic events or gates. In other words, a FTA is performed to determine the precise probability of a TE. A fault tree allows to determine the operational relationship among different components under different

modes to derive analytical expressions for the probability of failure. A general static FTA is mainly based on the graphical representation of different combinations of basic events which result in the TE. Since traditional logical gates like AND, OR, etc., are not capable of demonstrating the dynamic behavior of failure mechanisms in a system with dependent events and failures, a DFTA is used [4,5]. The DFTA employs dynamic gates in its FT to represent a sequential notation for the system failures to consider their order and their combinations. By considering the order of failures in DFTA, the dynamic relationships between the TE and the basic events can be analyzed, whereas the traditional static FT only expresses the occurrence of basic events. It should be noted that solving a fault tree requires proper methods to obtain the probability of occurrence of the TE based on the occurrence probability of the basic events. A DFT has at least one dynamic gate along with a combination of static gates such as AND ($\cdot$), OR ($+$), and voting gates. The most commonly used dynamic gates in DFTA are namely; the priority AND (PAND) gate, the sequence enforcing (SEQ) gate, the standby or spare (Spare) gate, and the functional dependency (FDEP) gate [5–7]. The dynamic gates in fault trees provide a conceptually simple modeling framework to represent system-level reliability in terms of interactions between component reliabilities through basic events relationships [8–11].

1.1 Problem statement

The weakness of the static FTA necessitates a more dynamic mechanism for analizing complicated and critical cases, such as avionic systems in an aircraft. Over the last decades, there have been many airplane crashes reported due to unforeseen failures in different aircraft sections, particularly, in flight control hardware and software systems in airplanes. Therefore, the use of more accurate FT analyses which dynamically predict different faults in different components of avionic systems of planes are indispensable. The use of dynamic gates enables the analysis to be clearer and more accurate compared to conventional FTA. Designing a reliable DFT for an aircraft that considers failures in the most significant flight components, either the mechanical structures or computer-based systems of the plane, allows for a high level of safety. However, solving such a fault tree requires precise algorithms to determine the failure probability of the TE based on the relationships between the basic events.

1.2 Objectives of the project

The DFTA, should be designed such that it shows the total failure of an aircraft based on the states of the basic events and their relationships. In this regard, the main questions are:

- How can we estimate the safety of an avionic system in an aircraft by using DFTA?
- Can we dynamically model the faults in different mechanical components, such as Elevator, Rudder, Aileron, and computer-based systems like flight control computers of the plane?
- Which dynamic gates should we use to relate the failures in each component of the DFT in order to estimate the total failure probability of it precisely?

Based on these main questions, the specific objectives of this project can be given as follow:

1. Develop a mathematical approach to determine the behavioral models or structure functions of the dynamic gates and the subtrees in the DFT of an aircraft.
2. Algebraically determine the probabilistic models of dynamic gates and subtrees in the designed DFT.
3. Obtain the total failure probability of the designed DFT through algebraic analysis of the subtrees in the structure and their individual failure probabilities.

1.3 Methodology

Aside from an ongoing literature review, this research involves four main activities. Phase I is allocated to investigation of the defects in the different mechanical structures and flight control computers of an aircraft. In Phase II, the failures in each component are considered as basic events in the corresponding dynamic gates (subtrees) to algebraically determine the behavioral model of the gates. The failure probabilities of the designed dynamic gates are investigated mathematically in Phase III. Finally, in Phase IV, the TE which is the total failure of the avionic system of the aircraft is calculated using a precise algebraic approach that relates the failure probabilities of the subtrees.

Defects investigation in mechanical components and flight control computers (Phase I)

An aircraft has several mechanical components which play key roles in controlling its flight, such as Elevators, Rudder, and Ailerons to perform pitch control, roll control and yaw control, respectively. Additionally, there are three flight control computers in the aircraft functioning one at a time. The failures in these components should be initially well investigated to ensure modeling them by appropriate dynamic gates.

Obtaining the behavioral models of the dynamic gates (Phase II)

The failures in each of the flight components in the aircraft are used as basic events in the dynamic gate as a subtree of the designed DFT. Using temporal operators and mathematical theorems, the behavioral model or structure function of each dynamic gate is calculated, mathematically.

Algebraic determination of the failure probability of each subtree (Phase III)

Depending on the type of dynamic gates used for modeling the failure in a flight component, different methods should be used to obtain the failure probability of the corresponding subtree in the DFT. The reason lies in the fact that the behavioral model and probabilistic model of each dynamic gate is different from the others.

Algebraic determination of the total failure of the aircraft (Phase IV)

The total failure probability of the aircraft, which is represented as the TE in the designed DFT, is determined by a precise mathematical algorithm which relates the failures in the subtrees.

CHAPTER 2 LITERATURE REVIEW

Fault Tree Analysis (FTA), as one of the most commonly used methods in industry, has gained a great deal of attention in many industrial applications where quantitative reliability and safety analysis are crucial. It makes use of logical and probabilistic techniques to analyze all realistic ways of occurrence for an undesired state called Top Event (TE) in a given system. Faults can originate from several sources of failure, such as component hardware, human operations, software, etc. Such an analysis can be summarized by a graphical representation called a Fault Tree (FT) and that describes the logical interrelationships between the basic failures which causes the undesired TE. A fault tree is represented by basic events and gates connecting the events to each other. The top gate describes the given system failure as TE, which is connected to the lower gates (basic events) in a tree shape network. Basic events cause basic failures in the system. The top event is analyzed throughout the fault tree using its constituent basic events or gates. Using this model, FTA is performed to determine the precise probability of the TE. However, it is an expensive process to develop the FT model and solve it accurately for any complex and large-scale FT representing a large industrial systems, such as a nuclear, oil and gas plant, or an airplane.

In a FT, a sequential failure analysis is performed in which an undesired state of a system i.e., the TE, is analyzed by combining the failures in a series of lower-level basic events. The FT is used to model the probability a system fails based on the probability failures of individual basic components. It allows to determine the operational relationship among different components under different operation modes, and the fault tree is used to derive analytical expressions for the probability of failure.

2.1 Structural representation of FTA and its benefits

In FTs, two types of analysis can be generally conducted:

1. A qualitative analysis performed by using Minimal Cut Sequences (MCS).
2. A quantitative analysis used to calculate of absolute probabilities.

The main benefits of constructing a fault tree diagram are:

- Explicitly shows relationships necessary to result in the fault or failure.
- Highlights critical elements related to system failure.

- Creates a visual aid for system analysis and management. Apparently, most managers prefer graphics to text.
- Provides an effective way to analyze the system.
- Exposes system behavior and possible interactions. FTA diagram allows for examination of ways that a fault may occur which may expose Non-obvious path to failure that analysis approaches miss.
- Implements method thet considers human errors.
- Promotes effective information communication. Such a diagram visually presents information on system analysis in a clear and concise way.

Fig. 2.1 shows different symbols used to represent fault trees in these analysis. These symbols are mainly categorized in two main groups called Gate Symbols and Event Symbols. Event symbols are used to represent either Primary Events or Intermediate Events. More explanations about the symbols which are used in this research will be given in the reset of the thesis.

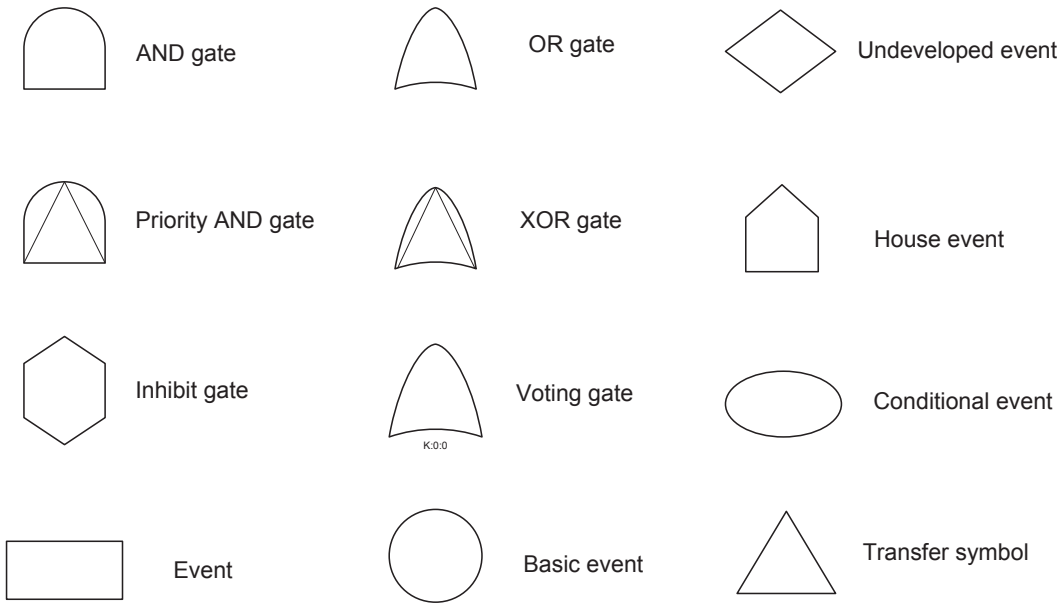


Figure 2.1 Schematic of fault tree symbols.

The main goal of using FTA is to identify potential causes of a given system failures and to evaluate the probability of the top event in the system. A unique set of events that cause

the top event to occur is called cut sequence (subtree). Figure 2.2 depicts and example of different cut sequences resulting in the TE. As can be seen in this figure, for the TE the cut sequences are the gates containing basic events (A,D), (B,D), (C,D), and (G,H) or the events (E), and (F).

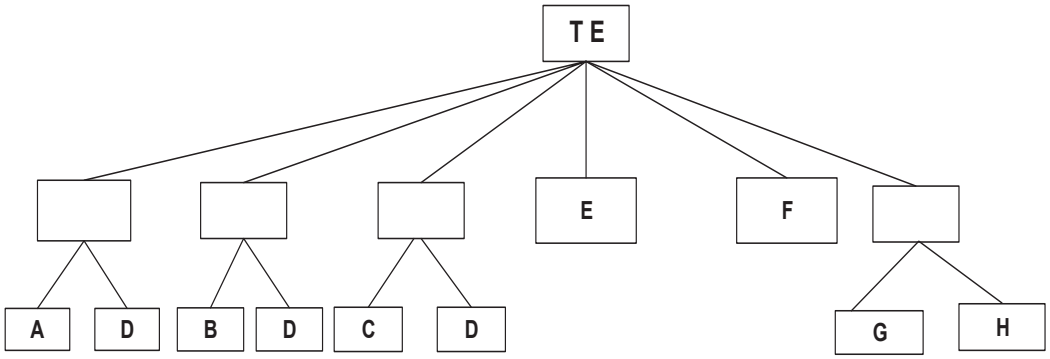


Figure 2.2 Schematic of different cut sequences for a TE in FTA.

Compared to other formal safety analysis methods, formal FTA is the only one which has a human readable and understandable logic background structure. However, there are some limitations to FTA for analyzing specific events. For instance, in some applications, conventional FTA may lead to very large trees if the analysis is extended in depth. Furthermore, it is difficult to apply it to systems with partial success. Another shortcoming is that it can be costly in both time and effort. As a result, a dynamic analysis of fault trees should be employed to address these drawbacks in the conventional FTA thanks to its dynamic property to consider the order of basic events occurrence.

FTA is mainly based on the graphical representation of different combinations of basic events which result in the TE. In this analysis, all possible ways for the TE occurrence are logically investigated by considering all the events statistically independent, and their relationships are represented by logical Boolean gates. Traditional logical gates like AND, OR, and Voting gates, are not able to demonstrate the dynamic behavior of failure mechanisms in a given system that has sequence dependent events and failures with spares and dynamic redundancy management. To circumvent this issue, DFTA can be used as it has dynamic gates in its FT to represent sequential events leading to system failures considering their order and also their combinations [5, 8, 12]. DFTA has been proposed as a method to ensure high safety [8]. It should be underlined that solving a fault tree requires proper methods to obtain the probability of occurrence of the TE on the basis of the occurrence probability of the basic events. Traditional static FTs has been solved by using various techniques like

disjoint products, inclusion/exclusion technique, combinatorial approach suitable for static phased, mission systems, etc. [13]. In such scenarios, as the system possesses several types of dynamic metrics, the FT becomes larger, and thus, it requires a more complex analysis. In this sense, DFT analysis can be exploited to model the dynamic behavior of the system to be analyzed to reduce the computational and time costs [5]. In DFT analysis, in addition to the combination of failure events, their order of occurrence is considered. In DFT, considering the order of failures in a Boolean model, allows for analysing the dynamic relationships between the TE and the basic events which cannot be carried out by using traditional static FT that only expresses occurrence of basic events. DFT can dynamically define various failure sequences and functional dependences using special dynamic gates such as the priority AND (PAND) gate, the sequence enforcing (SEQ) gate, the standby or spare (Spare) gate, and the functional dependency (FDEP) gate [5]. A DFT has at least one dynamic gate along with a combination of static gates such as AND, OR, and voting gates.

There have been many research activities conducted on methods of solving DFTs [5,8,12–21]. For instance, modularization method allows for classifying different independent sub-trees using dynamic gates so that different Markov models can be applied to each of them [5,17,18]. However, the dark point of using Markov model is that the number of system states increases exponentially the system size and its complexity. As a result, the Markov model-based solution if needed is generally integrated with the so-called combinatorial solutions [22]. The method of integrating Markov solution with combinatorial solutions is called modularization. Modularization is the method of finding independent components in a FT and solving the static gates using combinatorial techniques and dynamic gates with Markov solution and integrating the two methods to determine the system reliability [23,24]. However, as mentioned earlier, in systems with large number of basic events, the method suffers from the constraint in state space and the required high computational time. To address this issue, it is essential to minimize the state space by further modularization of the dynamic sub-trees [19]. The numerical technique proposed in [8] for solving dynamic gates, although effective to minimize the state space, cannot be used for repairable systems. In [20, 21] the presented Bayesian network-based technique was used to solve the dynamic FTs with minimum state space.

CHAPTER 3 DYNAMIC FAULT TREE ANALYSIS

For a large number of input gates which is common in many nuclear power plants where the probabilistic safety assessment is vital, using Markov models for solving the dynamic gates will impose a large calculation work load. Additionally, it is not trivial to use a Markov model for modeling systems whose failure and repair distribution functions are exponential. Furthermore, some scenarios may be challenging to be solved using analytical methods, and thus, it is essential to employ a more appropriate technique of modeling to implement the required dynamic gates for their FTA. One way to tackle this problem is to use Monte Carlo simulation-based technique to implement dynamic gates of the DFTA that can simulate the actual process and random behavior of the system in a way that uncertainty in reliability modeling is eliminated [25, 26].

3.1 Markov analysis

Markov analysis is used to model systems with many different states ranging from a *perfect function* state to a *total fault* state. It is well suited for modeling the reliability characteristics of a system particularly a small system which has complex maintenance strategies. In this analysis, the possible transition between different states is described by a so-called Markov-model diagram. Markov analysis allows for estimating the average time that the system is in each state which can be used to form a base for the metrics of interest largely employed in economic modeling. Additionally, the model can be used for estimating the average frequency that the system visits different states which might be used for estimating the need for spare parts, and maintenance process. It can also provide an estimation of the mean time until the system migrates to a specific state, e.g., a critical state.

The procedure in Markov analysis starts with sketching the system, defining the states, and drawing the Markov diagram with the corresponding transition rates. By doing the required quantitative assessment one can carry on the process to present the results of interest from the analysis. Since markov analysis is not used in this research work, we just give an example of it and more details about the method will be out of the scope of this thesis.

3.1.1 An example for Markov analysis

Considering that a power plant has a main engine working actively and also a spare engine which is in standby as long as the active one works well as shown in Fig. 3.1.

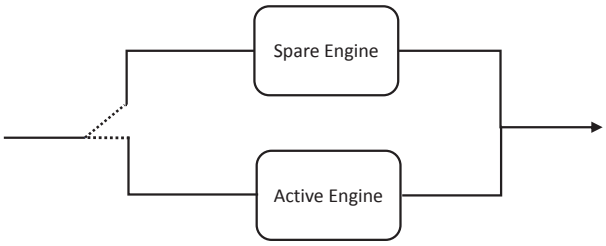


Figure 3.1 System schematic of a power plant with one active engine and one spare engine.

The system states can then be defined as follows

- S_1: active engine state
- S_2: standby engine state
- $S_i = \begin{cases} 1 \; if \; the \; i^{th} \; engine \; is \; functioning, \\ 0 \; if \; the \; i^{th} \; engine \; fails. \end{cases}$

Assuming that the standby engine will function as soon as the failure of the active one takes place, and in case both engines fail they will be repaired by maintenance personnel immediately, the state transitions for this system can be expressed as the followings

- λ_1: Active engine failure rate
- λ_2: Standby engine failure rate if it has been functioning. λ_2=0 in standby state.
- μ_1: Active engine repair rate. $1/\mu_1$= Mean downtime when the active engine is in fault state.
- μ_{Both}: Repair rate when both engines are in failure state. If the active engine had failed first and then the spare engine started to work and failed later, any of them can start to work again with the repair rate μ_{Both}.

In Markov state space diagram, each state is represented by a circle and the transition rates between the different states are shown by arrows. The Markov diagram and the description of its different states demonstrates the total qualitative representation of the system. In this example, the system state can be presented as follows

- System state = 2, if both engines are not in fault state ($S_1 = S_2 = 1$),

- System state = 1, if the active engine has failed ($S_1 = 1$) and the spare engine is functioning ($S_2 = 0$),
- System state = 0, if both engines are in fault state ($S_1 = S_2 = 0$),

Therefore, the Markov diagram for this system can be represented by

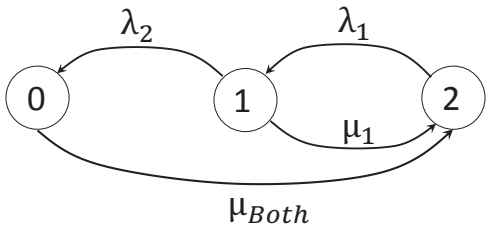


Figure 3.2 Markov diagram for the power plant example with one active engine and one spare engine.

Depending on the quantities of interest, the analysis can be done by finding the required transition matrix. The transition matrix elements $a_{i,j}$ show the transition rates between the states from i state to j state. The diagonal elements are determined at the end in a way they fulfil the condition that all cells in a row adds up to zero. The transition matrix for this example can be given by

$$A = \begin{bmatrix} -\mu_{Both} & 0 & \mu_{Both} \\ \lambda_2 & -\lambda_2 - \mu_1 & \mu_1 \\ 0 & \lambda_1 & -\lambda_1 \end{bmatrix}. \tag{3.1}$$

3.2 Monte Carlo simulation-based technique for dynamic gates

In this technique the actual process and random behavior of a system is simulated on a computer model to provide a realistic scenario of the system. In this method, the problem is treated as a series of actual experiments carried out within a simulated time. By counting the number of occurrences of an event within simulation time, Monte Carlo approach in DFTA provides an estimation of the probability and other important indices. To this end, it uses the probability density function of the time to failure and repair for all basic events. The events are simulated within a specific mission time to show the available duration (up) state and unavailable duration (down) state. A down state can be caused by an unexpected failure and it can be recovered depending on its repair action time. The up and down state

durations are random depending on the probability density function (*pdf*) of time to failure and time to repair, respectively. The states change with time alternatively can be depicted in the so-called state-time diagram. To determine the time to failure or the time to repair for a state-time diagram of a random variable t that follows an exponential distribution with the failure rate of λ, the *pdf* and the cumulative density function $F(t)$ are given by

$$f(t) = \lambda e^{-\lambda t}, \tag{3.2}$$

$$F(t) = \int_0^t f(t)dt = 1 - e^{-\lambda t}. \tag{3.3}$$

By solving for t, one can calculate the time to failure for a given $F(t)$ and λ as the following

$$t_{failure} = \frac{1}{\lambda} \ln(\frac{1}{1 - F(t)}). \tag{3.4}$$

Similarly, the time to repair can be determined by solving for t using equation (3.3) as follow

$$R(t) = 1 - F(t) = e^{(-\lambda t)}. \tag{3.5}$$

Therefore,

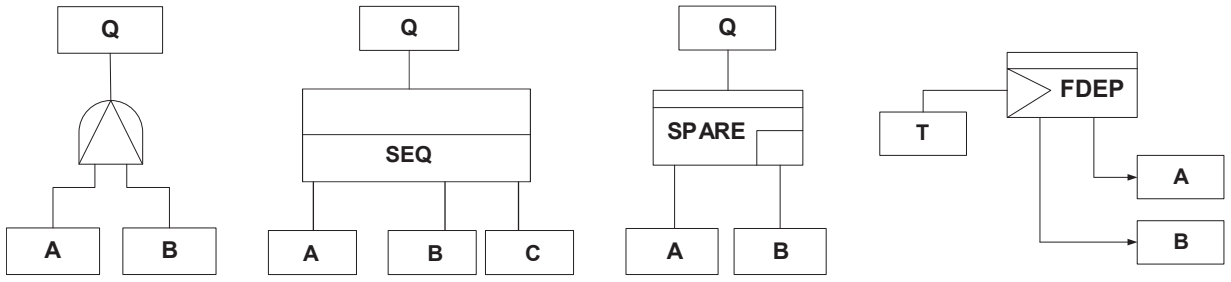
$$t_{repair} = \frac{1}{\lambda} \ln(\frac{1}{R(t)}). \tag{3.6}$$

3.3 Dynamic gates in DFTA

DFT can dynamically define various failure sequences and functional dependences using special dynamic gates. The most commonly used gates in DFTA are shown in figure 3.3 and they are namely; the priority AND (PAND) gate, the sequence enforcing (SEQ) gate, the standby or spare (Spare) gate, and the functional dependency (FDEP) gate [5].

3.3.1 The PAND gate in DFTA

The PAND gate is logically equivalent to the traditional AND of which the order of occurrence of its input events is of importance in the output event occurrence. The output of a PAND gate with two inputs becomes true (failure) if and only if both basic events A and B have reached failure simultaneously or A has failed before B. The time to failure and repair can be determined on the basis of the *pdf* of failure time and the *pdf* of repair time, respectively. The sequence of failure and repair states is continued until the predetermined system mission



(a) (b) (c) (d)

Figure 3.3 Schematic illustration of dynamic gates used in DFTA; (a) the priority AND (PAND) gate, (b) the functional dependency (FDEP) gate, (c) the standby or spare (Spare) gate, and (d) the functional dependency (FDEP) gate.

time. The state-time diagram for the PAND gate can be developed by comparing that of the two components (A and B). If and only if both A and B components have failed in a preassigned order (first A and then B), the output of the PAND gate becomes failure. Figure 3.4 summarizes various scenarios in a PAND gate depicting the failure and non-failure states of the gate.

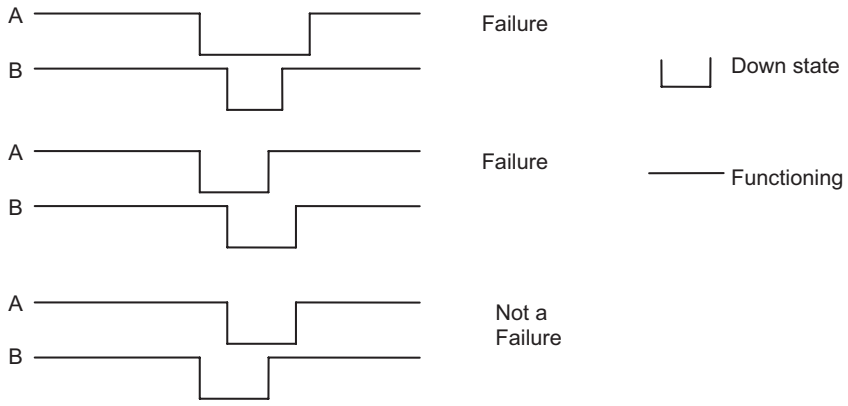


Figure 3.4 State-time diagram of gate PAND. A PAND gate fails if and only if both A and B fail, and A fails before B.

3.3.2 The SEQ gate in DFTA

The input events of an SEQ gate occur in a specific order. An SEQ gate in a DFT does not lead to failure when identified events occur in an order different from the preassigned

order. This gate is similar to the PAND gate, but it prevents the DFT from undefined failure sequences to happen and it is much stronger than a PAND gate. A PAND gate has a failure output if A fails before B. However, B may fail before A without generating a failure output. As shown in figure 3.5, when an SEQ gate has three inputs with repairable components to reflect the behavior of a system, the state time profile of the first component is generated based on its failure and repair rate. In this example, the mission time of the second component is the down time of the first one, and the mission time of the third component is the down time of the second one. This means, once the first component reaches failure, the second component starts to operate at its t=0. As a result, the time to failure for the second component (TTD2) and its time to repair which is the second component down time (CD2) will be generated. Similarly, the operation of the third component commences at its t=0, when the second one fails. Consequently, the time to failure for the third component (TTD3) and its time to repair which is the second component down time (CD3) will be generated. The downtime of the SEQ gate is when the three components are down simultaneously. This process is happened again for all down states of the first component shown in figure 3.5.

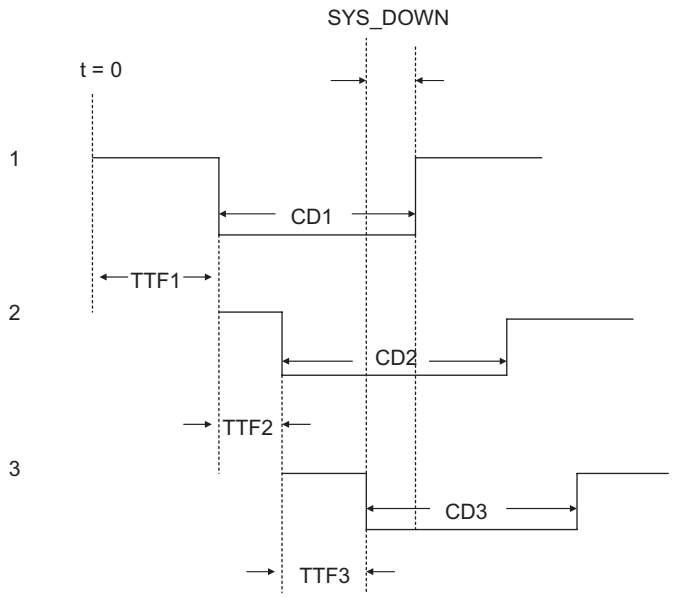


Figure 3.5 State-time diagram of gate SEQ. TTFi denotes time to failure for ith event, CDi represents component down time for the ith event, and SYS_ DOWN is system down time.

3.3.3 The Spare gate in DFTA

A Spare gate defines one or more main components that can be replaced by their spares with the same functionality. This gate will generate a failure output if the total number of operationally powered spares and principal components is less than a threshold (required minimum value). Additionally, Spare gates generate failure if they are unpowered (dormant), but the rate of failure for an unpowered spare ($\alpha\lambda$) is less than that of the powered spare (λ), where α is the dormancy factor such that $0 \leq \alpha \leq 1$ and λ represents failure rate. Spare elements with $\alpha = 1$ and $\alpha = 0$ are called "*hot*" and "*cold*" components, respectively. Consider a Spare gate in which A is the active component and B is the remaining component in a Spare gate. The state-time diagrams of the components are then generated starting with the active component followed by the spare components in the order from left to right.

It is essential to note that in a Spare gate, the time to failures and the time to repairs of active components are alternatively generated based on the corresponding *pdf*s and the sequence will continue until their mission time starts. On the other hand, for the spare components as long as they are unpowered they are either in standby state or maybe in failure state. Additionally, a spare component can also be unavailable because of a scheduled activity that requires the component to be in test state or maintenance state. In this case, the component has multi-states and thus, its operation should be modeled precisely to cope with different states. As a result, in the state-time diagrams, the down times should be considered first due to the scheduled test and maintenance policies. The test override probability should also be considered to ensure the component availability during testing. It is notable that a failure during a standby period cannot be recognized until its testing starts and thus, the time from this failure to its identification must be considered as down time. The next metrics are the standby downtimes which are determined by the *pdf* of the standby time to failure and the *pdf* of the standby time to repair. Furthermore, it is of great importance to ensure that not only the availability of the standby component is on demand but also it is capable to succeed its mission. To this end, the time to failure is calculated based on the operating failure *pdf* and is checked with the mission time (the down time of the active component). When the first standby component fails prior to recovery of the active component, the next spare component will receive the demand.

Figure 3.6 depicts different scenarios of a Spare gate. As can be seen in the first scenario, even if the demand due to the failure of the active component A is met by the standby component, the Spare gate may reach failure since the standby component has failed before the recovery of the active component. According to the second scenario shown in this figure, the standby component has met the demand due to the failure of the active component and

it failed while it was in its dormant mode, i.e., before and after the failure of the active component. Here, the failures of the standby component have no effect on the success of the gate. In the third scenario, the Spare gate experiences failure since the demand came in when the standby component has already been in failure mode, however, the overall downtime is reduced because of the recovery of the standby component before that of the active one.

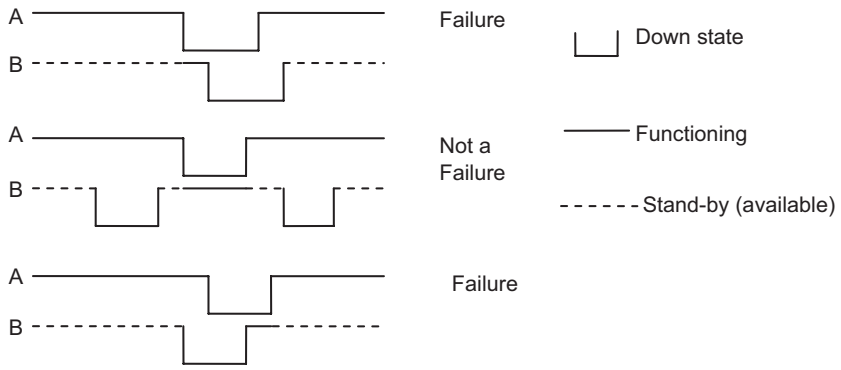


Figure 3.6 State-time diagram of gate Spare.

3.3.4 The FDEP gate in DFTA

In the FDEP gate, one of the inputs called trigger input can be either a basic event or the output of another gate in the FT, while the rest of the inputs are functionally dependent on the trigger event. In other words, the dependent basic events occur by the force of the trigger input which means the separate occurrence of any dependent basic event will not affect the trigger event. Once the trigger event (T) takes place in the gate, the dependent events (A and B) of the gate occur. In this gate, the failure and repair times are generated according to the *pdf* of the trigger event. As can be seen in the first scenario of figure 3.7, the dependent events are in virtual failure mode but functioning during the failure (down) time of the trigger event. It can be inferred from the second scenario of the figure that if a dependent event occurs individually, it will not affect the trigger event.

In [27–30] it was shown that in systems with perfect fault coverage, the FDEP gate can be used to perform as an OR gate. However, in redundant systems that cannot have perfect coverage, undetected faults cannot be covered and as a result they can propagate in the whole system which may lead to system failure [31]. To address this problem for systems with imperfect fault coverage, a complex method based on FDEP gates was implemented in [31] for an efficient reliability analysis. In [11], an inclusion/exclusion methods with PAND

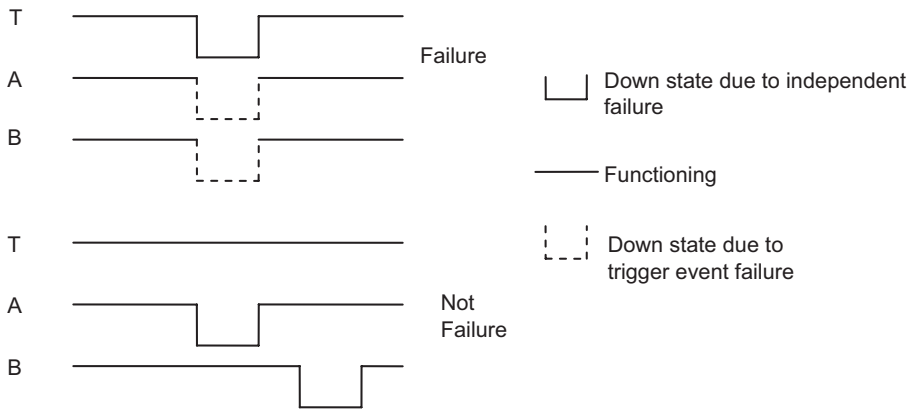


Figure 3.7 State-time diagram of gate FDEP.

gates were used for an exact analysis of DFT with repeated events. However, the method can only be used for a system with exponentially distributed failures whose minimal cut set information is available for the analysis.

3.4 Temporal operators and theorems in DFTA

The behavioral models of dynamic gates in a DFT is defined by employing temporal operators which model the order of occurrence of events. These temporal operators are namely; Non-inclusive BEFORE, SIMULTANEOUS and Inclusive BEFORE. The Non-inclusive BEFORE is represented by $\lhd$ and used for modeling dynamic gates such that the structure function of a DFT can be determined. The $a \lhd b$ occurs if event a occurs before event b, or if a occurs and b never occurs. The latter happens when $b \equiv \perp$ which results in $a \lhd \perp = a$, where $\perp$ denotes null(nothing) with a failure time $t_\perp \to \infty$ [32]. The formal definition of Non-inclusive BEFORE operator based on the time of occurrences of a and b (t_a and t_b) is expressed as

$$a \lhd b = \begin{cases} a & \text{if } t_a < t_b, \\ \perp & \text{if } t_a > t_b, \\ \perp & \text{if } t_a = t_b. \end{cases} \tag{3.7}$$

The SIMULTANEOUS operator is denoted by $\triangle$ and used for modeling simultaneous events. For two statistically independent basic events a and b it is notable that $a \triangle b = \perp$. According

to [32] the SIMULTANEOUS temporal operator can be defined as follow

$$a \triangle b = \begin{cases} \perp & \text{if } t_a < t_b, \\ \perp & \text{if } t_a > t_b, \\ a & \text{if } t_a = t_b. \end{cases} \tag{3.8}$$

The Inclusive BEFORE operator is expressed as $\trianglelefteq$ and exploited to model an alternative inclusive version of dynamic gates in DFTA. It should be noted that $a \trianglelefteq b$ occurs if a occurs before b ($a \triangleleft b$) or if a and b occur simultaneously ($a \triangle b$) [32, 33]. Therefore,

$$a \trianglelefteq b = \begin{cases} a & \text{if } t_a < t_b, \\ \perp & \text{if } t_a > t_b, \\ a & \text{if } t_a = t_b. \end{cases} \tag{3.9}$$

The three temporal operators can be used to define several useful theorems to structurally develop and simplify DFTs [32]. Some useful theorems based on the temporal operators which are used in the rest of this research work are given below:

$$a \triangleleft a = \perp, \tag{3.10}$$

$$\perp \triangleleft a = \perp, \tag{3.11}$$

$$a \triangleleft \perp = a, \tag{3.12}$$

$$a + (a \triangleleft b) = a, \tag{3.13}$$

$$(a \triangleleft b) + b = a + b, \tag{3.14}$$

$$a \cdot (a \triangleleft b) = a \triangleleft b, \tag{3.15}$$

$$a \triangle a = a, \tag{3.16}$$

$$a \triangle \perp = \perp, \tag{3.17}$$

$$a \triangle b = b \triangle a, \tag{3.18}$$

$$a \triangle (b \triangleleft c) = (a \triangle b) \cdot (b \triangleleft c), \tag{3.19}$$

$$a \triangle (b \trianglelefteq c) = (a \triangle b) \cdot (b \trianglelefteq c), \tag{3.20}$$

$$a + (a \triangle b) = a, \tag{3.21}$$

$$a \cdot (a \triangle b) = a \triangle b, \tag{3.22}$$

$$a \trianglelefteq a = a, \tag{3.23}$$

$$\perp \trianglelefteq a = \perp, \tag{3.24}$$

$$a \trianglelefteq \perp = a, \tag{3.25}$$

$$a + (a \trianglelefteq b) = a, \tag{3.26}$$

$$b + (a \trianglelefteq b) = a + b, \tag{3.27}$$

$$a \cdot (a \trianglelefteq b) = a \trianglelefteq b, \tag{3.28}$$

$$a \trianglelefteq b = a \triangleleft b + a \triangle b, \tag{3.29}$$

$$(a \cdot b) \trianglelefteq c = (a \trianglelefteq c) \cdot (b \trianglelefteq c), \tag{3.30}$$

$$(a \trianglelefteq b) \trianglelefteq c = (a \trianglelefteq b) \cdot (a \trianglelefteq c), \tag{3.31}$$

$$(a \trianglelefteq b) \cdot (b \trianglelefteq c) \cdot (a \trianglelefteq c) = (a \trianglelefteq b) \cdot (b \trianglelefteq c), \tag{3.32}$$

$$(a \triangleleft b) + (a \triangle b) + (a \cdot (b \triangleleft a)) = a. \tag{3.33}$$

3.5 Behavioral model of dynamic gates

The behavioral model of each dynamic gate can be determined using the temporal operators, Non-inclusive BEFORE ($\triangleleft$), SIMULTANEOUS ($\triangle$) and Inclusive BEFORE ($\trianglelefteq$), along with static gates such as AND (.), OR (+), and Voting gates.

3.5.1 Behavioral model of the PAND gate

As shown in Fig. 3.8, if the basic events of the PAND gate (A and B) occur simultaneously or A fails before B, the output event Q takes place.

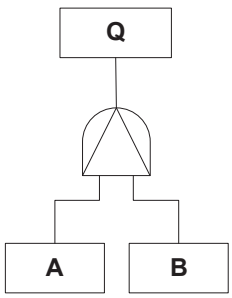


Figure 3.8 Schematic illustration of the PAND gate.

Therefore, the behavioral model of the PAND gate can be expressed as

$$Q = (A \cdot B) \cdot (A \trianglelefteq B). \tag{3.34}$$

Using the theorem given by equation (3.28), it can be simplified as follows:

$$Q = B \cdot (A \trianglelefteq B). \tag{3.35}$$

3.5.2 Behavioral model of the FDEP Gate

As depicted in Fig. 3.9, the two dependent basic events A and B of an FDEP gate may occur either by themselves or forced by the trigger event T.

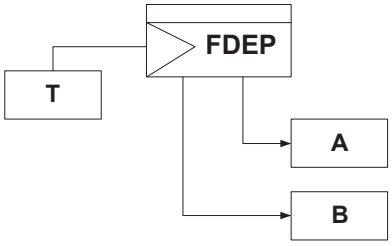


Figure 3.9 Schematic illustration of the FDEP gate.

In order to indicate the effect of the trigger event T on the behavior of basic events A and B, the substituted variables A_T and B_T is defined. In this sense, A_T denotes the case that the trigger event T forces the basic event A to fail or A fails by itself before failure of T, i.e., $(A \trianglelefteq T)$. Similarly, B_T represents the case that B fails by the force of T or by itself before T fails, i.e., $B \trianglelefteq T$.

$$FDEP : \begin{cases} A_T = T + (A \trianglelefteq T) = T + A \\ B_T = T + (B \trianglelefteq T) = T + B \end{cases} . \tag{3.36}$$

It can be inferred from equation (3.36) that the behavior of a dynamic FDEP gate is equivalent to the behavior of OR gate.

3.5.3 Behavioral model of the Spare gate

The common Spare gate has two basic events, i.e., the primary event A and the spare event B, which is also called single Spare gate, as shown in Fig. 3.10. The number of input events in a Spare gate can be more than two in an increasing order of complexity. Additionally, in some configurations, several Spare gates can share a single spare event [9]. In this thesis, the common Spare gate with two input events is the only configuration that is considered.

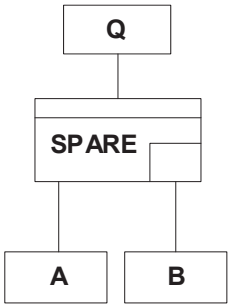


Figure 3.10 Schematic illustration of the Spare gate with two input events, the primary event A and the spare event B.

The output event Q of the Spare gate occurs when both the primary event A the spare event B fail such that A and B do not fail simultaneously. The reason lies in the fact that A and B are two independent basic events and thus, $\mathrm{A} \triangle B = \perp$. This means Q occurs if A fails either before B or after B. The scenario of B fails after A occurs when B is in active mode (B_a), while failure of B before A takes place when B is in its dormant mode (B_d). As a result, in quantitative analysis of DFTs, the failure distribution of B when it fails during its active mode is different from that of its failure in dormant mode [9]. It should be noted that B cannot be in active mode and dormant mode simultaneously, and thus, $B_a \cdot B_d = \perp$. Therefore, behavioral model of the Spare gate can be expressed as

$$Q = B_a \cdot (A \lhd B_a) + A \cdot (B_d \lhd A). \tag{3.37}$$

3.6 Probabilistic model of dynamic gates

This section is allocated to the determination of the probabilistic model of the dynamic gates, PAND, FDEP, and Spare, using the corresponding behavioral model presented in Section 3.5. In this sense, the behavioral model of these dynamic gates are exploited to define the structure function of a given DFT from which probabilistic model of dynamic gates can be obtained. The probabilistic model of the dynamic gates allows to determine the failure probability of the TE of the DFT. For an exponential distribution, the probability density function (*pdf*) represented by $f(t)$ and the cumulative distribution function (*cdf*) denoted by $F(t)$. Knowing that $f(t)$ is equal to the derivative of $F(t)$, i.e., $f(t) = F'(t)$, they are given by

$$f(t) = \lambda e^{-\lambda t}, \tag{3.38}$$

and

$$F(t) = \int_0^t f(t)dt = 1 - e^{-\lambda t}. \tag{3.39}$$

According to [8, 34], for two basic events a and b the following expressions hold:

$$\begin{aligned} Pr\{a+b\}(t) &= F_a(t) + F_b(t) - (F_a(t) \times F_b(t)) \\ &= 1 - \Big(1 - Pr\{a(t)\}\Big)\Big(1 - Pr\{b(t)\}\Big), \end{aligned} \tag{3.40}$$

$$Pr\{a \cdot b\}(t) = F_a(t) \times F_b(t), \tag{3.41}$$

$$Pr\{a \lhd b\}(t) = \int_0^t f_a(u)\,(1 - F_b(u))\,du, \tag{3.42}$$

$$\begin{aligned} Pr\{b \cdot (a \lhd b)\}(t) = Pr\{[a,b]\}(t) &= \int_0^t f_b(u)\,\Big(\int_0^u f_a(v)\,dv\Big)\,du \\ &= \int_0^t f_b(u)\,F_a(u)\,du. \end{aligned} \tag{3.43}$$

Equation (3.42) can be proven by simplifying equation (3.33) by using equation (3.43) and the fact that the basic events are statistically independent ($a \triangle b = \perp$). Thus,

$$(a \lhd b) + (a \triangle b) + (a \cdot (b \lhd a)) = (a \lhd b) + (a \cdot (b \lhd a)) = a, \tag{3.44}$$

$$Pr\{(a \lhd b)\}(t) + Pr\{(a \cdot (b \lhd a))\}(t) = Pr\{(a \lhd b)\}(t) + \int_0^t f_a(u)\,F_b(u)\,du = Pr\{a\}(t), \tag{3.45}$$

which results in

$$Pr\{(a \lhd b)\}(t)) = Pr\{a\}(t) - \int_0^t f_a(u)\,F_b(u)\,du, \tag{3.46}$$

which can be written as

$$Pr\{(a \lhd b)\}(t)) = F_a(t) - \int_0^t f_a(u)\,F_b(u)\,du. \tag{3.47}$$

Since, $f(t) = F'(t)$, the derivative of the both sides of equation (3.47) leads to

$$\begin{aligned} \frac{d(Pr\{(a \lhd b)\}(t))}{dt} &= f_a(t) - f_a(t)\,F_b(t) \\ &= f_a(t)(1 - F_b(t)), \end{aligned} \tag{3.48}$$

which can be used to solve for $Pr\{(a \lhd b)\}(t)$ as follow

$$Pr\{(a \lhd b)\}(t) = \int_0^t f_a(u)\,(1 - F_b(u))\,du. \tag{3.49}$$

In the next section, using these probabilistic expressions, the probabilistic models of gates PAND, FDEP, and Spare are defined from their behavioral model.

3.6.1 Probabilistic model of the PAND gate

As explained in Section 3.5.1, the behavioral model of gate PAND is defined as $Q = B{\cdot}(A \trianglelefteq B)$. Using equation (3.29), the output of the PAND gate can be developed as

$$\begin{aligned} Q &= B \cdot (A \lhd B + A \triangle B) \\ &= B \cdot (A \lhd B) + B \cdot (A \triangle B), \end{aligned} \tag{3.50}$$

and according to equation (3.18), and equation (3.22), it can be expressed as

$$Q = B \cdot (A \lhd B) + (A \triangle B). \tag{3.51}$$

Therefore, the behavioral model of gate PAND with two independent input events A and B i.e., $A \triangle B = \perp$ is simplified to

$$Q = B \cdot (A \lhd B). \tag{3.52}$$

Therefore, using equations (3.38), (3.39) and (3.43), the probabilistic model of the PAND gate can be given by

$$\begin{aligned} Pr\{Q(t)\} &= Pr\{B \cdot (A \lhd B)\}(t) \\ &= \int_0^t f_B(u)\,F_A(u)\,du \\ &= \int_0^t f_B(u) \int_0^u f_A(v)\,dv\,du \\ &= \int_0^t \lambda_B\; e^{-\lambda_B\, u}(1 - e^{-\lambda_A\, u})\,du \\ &= \lambda_B \int_0^t\; e^{-\lambda_B\, u}\,du - \lambda_B \int_0^t e^{-(\lambda_A+\lambda_B)\, u}\,du \\ &= 1 - \frac{\lambda_B}{\lambda_A + \lambda_B}(1 - e^{-(\lambda_A+\lambda_B)\, t}) - e^{-\lambda_B\, t} \\ &= \frac{\lambda_A}{\lambda_A + \lambda_B} - \frac{\lambda_B}{\lambda_A + \lambda_B} e^{-(\lambda_A+\lambda_B)\, t} - e^{-\lambda_B\, t} \end{aligned} \tag{3.53}$$

3.6.2 Probabilistic model of the FDEP gate

As explained in Section 3.5.2, the behavioral model of gate FDEP is given by equation (3.36). Using equation (3.40) the probabilistic model of gate FDEP can be obtained as follow

$$Pr\{Q(t)\} = \begin{cases} Pr\{A_T\} = Pr\{T + A\} \\ \qquad\qquad = F_A(t) + F_T(t) - (F_A(t) \times F_T(t)), \\ Pr\{B_T\} = Pr\{T + B\} \\ \qquad\qquad = F_B(t) + F_T(t) - (F_B(t) \times F_T(t)). \end{cases} \tag{3.54}$$

3.6.3 Probabilistic model of the Spare gate

The probabilistic model of a single Spare gate with one primary event A and one spare event B is based on the failure distribution of the spare event in its active and dormant mode. As long as B is in dormant mode, its failure distribution does not depend on A, and thus its *cdf* and *pdf* of B_d are functions of time represented by $F_{B_d}(t)$ and $f_{B_d}(t)$, respectively. On the other hand, when B is in active mode at the failure date of A (t_A), the failure distribution of the spare event B depends on A. In this scenario, the *cdf* and *pdf* of B_d depend on both time t and t_A [6].

The *cdf* and *pdf* of the primary event A with a failure rate λ_A can be expressed as

$$\begin{cases} F_A(t) = 1 - e^{-\lambda_A t} \\ \qquad\qquad\qquad \text{for } t \geq 0\,. \\ f_A(t) = \lambda_A e^{-\lambda_A t} \end{cases} \tag{3.55}$$

Similarly, the spare event B with a failure rate λ_B and a dormancy factor α has a *cdf* and a *pdf* given by

$$\begin{cases} F_{B_d}(t) = 1 - e^{-\alpha\lambda_B t} \\ \qquad\qquad\qquad \text{for } t \geq 0\,. \\ f_{B_d}(t) = \alpha\lambda_B e^{-\alpha\lambda_B t} \end{cases} \tag{3.56}$$

In the case of B_a, the *cdf* is exponential with its failure rate λ_B, and since B starts to be in its active mode at the failure date of A (t_A), it is continuous with the *cdf* of B_d at $t = t_A$, this requirement necessitates to define F_{B_a} as a function of (t, t_A) which can be expressed as

$$F_{B_a}(t, t_A) = 1 - e^{-\lambda_B(t - x(t_A))} \tag{3.57}$$

where x is a function of time t_a which can be determined by using the continuity criterion ($F_{B_A}(t, t_A) = F_{B_d}(t_A)$ at $t = t_A$) as follows

$$1 - e^{-\lambda_B(t_A - x(t_A))} = 1 - e^{-\alpha\lambda_B t_A}, \tag{3.58}$$

which results in

$$t_A - x(t_A) = \alpha t_A, \tag{3.59}$$

and thus,

$$x(t_A) = (1 - \alpha)t_A. \tag{3.60}$$

Consequently, the *cdf* and *pdf* of B_d can be given by

$$\begin{cases} F_{B_a}(t, t_A) = 1 - e^{-\lambda_B\,(t-(1-\alpha)\,t_A} \\ \\ f_{B_a}(t, t_A) = \lambda_B\, e^{-\lambda_B\,(t-(1-\alpha)\,t_A} \end{cases} \quad \text{for } t \geq (1-\alpha)t_A\,. \tag{3.61}$$

3.7 Conclusion

FT provide a conceptually simple modeling framework to represent system-level reliability in terms of interactions between component reliabilities. The weakness of the formal FTA modeling technique is that it necessitates a more dynamic technique for more complicated systems. In such dynamic FTA the use of dynamic gates empowers the analysis to be clearer and more reliable compared to the formal FTA. These dynamic gates are categorized in two groups based on their functionality. The first group consists of the PAND and FDEP gates which are priority dynamic gates, while the second group contains Spare and SEQ gates as their operations are based on the duration of failure events [6].

In a DFTA, the failure of a system depends upon the states of basic events of the dynamic gates and their relationships. The relationships originate from the system topology that is represented by a DFT using both dynamic and static gates.

CHAPTER 4 ALGEBRAIC DFTA FOR AVIONIC SYSTEMS

4.1 Introduction

Safe air transportation is the main goal of aviation organizations, such as International Civil Aviation Organization (ICAO) and Canadian Aviation Regulations (CARs) [35, 36] which are in charge of flight safety and the development of civil aviation. ICAO has defined specific standards and provisions, such as Standard and Recommended Practices (SARPs), Procedures for Air Navigation Services (PANS), Regional Supplementary Procedures (SUPPs) and guidance material. They are used to cover different technical and operational requirements which should be critically examined. These standards are exploited as benchmarks to analyze any failure in the avionic systems, and to prevent any chaos and hazard. There have been several methods to analyze such failures and errors, and this research work is allocated to the design and the investigation of a DFTA for an avionic system of aircrafts.

DFTA has gained a great deal of attention in many applications where quantitative reliability and safety analysis are crucial. It exploits logical and probabilistic analysis for all possible ways an undesired state called top event (TE) can occur in a given system. As the system possesses several types of dynamic metrics, the FT becomes larger and results in a more complex analysis. Dynamic FTA (DFTA) is used to model the dynamic behavior of a system. In addition to the combination of failure events in DFTA, their order of occurrence is also considered.

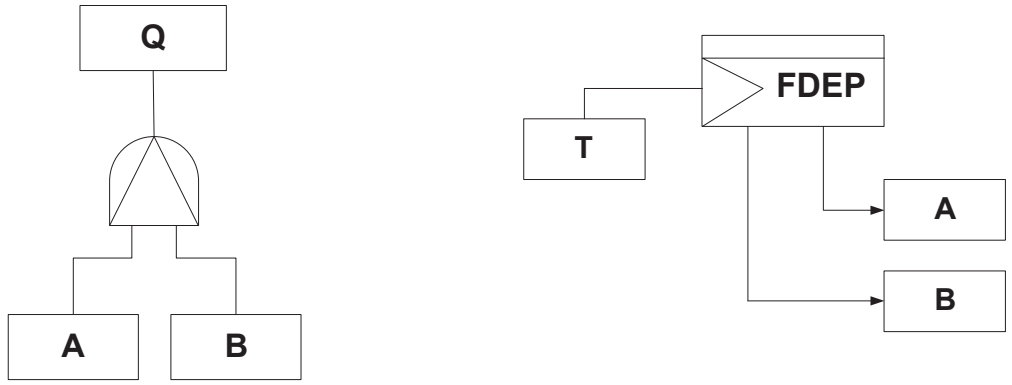
Much research has been conducted on methods of solving DFTs [12, 18, 19]. Markov analysis is one the most common methods for solving dynamic gates in DFT analyses. A limitation of this method is that it tends to generate a large number of system states, even for moderate size FTs, which increases the system complexity and the computing time of the analysis [19]. To circumvent this issue, Monte Carlo simulations are widely used to solve DFTs without relying on Markov states. In this technique, the actual process and random behavior of a system is simulated on a computer-based model to estimate the probability of occurrence of an event as a function of time. However, a simulation-based DFTA generally requires more computational time to achieve a high level of accuracy [18]. Compared to other methods, algebraic ones have proven to be promising candidates for solving dynamic fault trees as they are more straightforward and less computer intensive while providing a similar level of accuracy [8].

4.2 Theory of the algebraic model

Fault tree is used to model the probability of failure of a system based on subtrees of possible faulty events. Using FTA, the potential causes that can lead to system failures are identified and the probability of the TE is evaluated. Faults can originate from several sources, such as hardware components, human operations, software, etc. Such an analysis is represented by a graphical structure called a fault tree (FT) that describes the logical interrelationships between basic events causing the undesired TE. The FT allows to determine the operational relationship among different components under different modes to derive analytical expressions of the failure probability. A general static FTA is mainly based on the graphical representation of different combinations of basic events which result in the TE. In static analysis, all possible ways that can lead to the TE occurrence are logically investigated such that all events are considered statistically independent, and their relationships are defined by logical Boolean gates. Unlike traditional logical gates like AND, OR, etc., dynamic gates are capable to express dynamic behavior of failure mechanisms in a system with dependent events and failures. A DFTA uses dynamic gates to represent sequential events that can lead to system failure [5, 8, 12]. By considering the order of failures in DFTAs, the dynamic relationships between the TE and the basic events can also be analyzed, contrary to the traditional static FTs which only express the occurrence of basic events. DFTA has always been a reliable method for systems requiring a high level of safety [8].

4.2.1 PAND and FDEP dynamic gates

A typical DFT has at least one dynamic gate along with a combination of static gates, i.e., AND ($\cdot$), OR ($+$), and voting gates. In this work, we exploit two common types of dynamic gates; the priority AND (PAND) and the functional dependency (FDEP) gates shown in Fig. 4.1. The PAND gate depicted in Fig. 4.1a is logically similar to traditional AND but the occurrence order of its input events affects the occurrence of its output event Q. The output Q of a PAND gate with two inputs becomes true (failure state) if and only if both basic events A and B have reached failure such that A has failed before B. In the FDEP gate shown in Fig. 4.1b, basic events A and B may fail either by themselves, or due to the trigger event T. The dependent basic events occur as a consequence of the trigger input, which means that the individual occurrence of any dependent basic event will not affect the trigger event. Once the trigger event takes place in the gate, the dependent events (A and B) of the gate occur. The effect of trigger T can be modeled by using substituted variables A_T and B_T. In this regard, the basic event A_T fails if it is forced to fail by T. It may also fail by itself before failure of T.



(a) (b)

Figure 4.1 Schematic illustration of dynamic gates used in DFTA; (a) the priority AND (PAND) gate, and (b) the functional dependency (FDEP) gate with two basic events and one trigger event.

4.2.2 Structure function of PAND and FDEP gates

Using temporal operators, i.e., non-inclusive BEFORE ($\lhd$), SIMULTANOUS ($\triangle$) and Inclusive BEFORE ($\trianglelefteq$), for any non-repairable events, several theorems with their proofs are given in [6]. $a \lhd b$ occurs if event a occurs before event b, or if a occurs and b never occurs, i.e., $b \equiv \perp$, where $\perp$ denotes null (nothing). The operator $\triangle$ is used for modeling simultaneous events. For two statistically independent basic events a and b it is notable that $a \triangle b = \perp$. $a \trianglelefteq b$ occurs if a occurs before b ($a \lhd b$) or if a and b occur simultaneously ($a \triangle b$). According to [6] and using the theorems given by equations (3.28), (3.29), and (3.27), the behavioral models of the PAND and FDEP gates can be determined and simplified as follows

$$\begin{aligned} PAND:\ Q &= (A \cdot B) \cdot (A \trianglelefteq B) \\ &= B \cdot (A \trianglelefteq B) = B \cdot (A \lhd B + A \triangle B), \end{aligned} \tag{4.1}$$

$$FDEP: \begin{cases} A_T = T + (A \trianglelefteq T) = T + A, \\ B_T = T + (B \trianglelefteq T) = T + B. \end{cases} \tag{4.2}$$

4.2.3 Probabilistic expression of PAND and FDEP gates

The next step is to determine the probabilistic model of the dynamic gates used in this paper. Consider that the Cumulative Distribution Function (*cdf*) of an event x is represented by $F(x)$

and the Probability Density Function (*pdf*) of $f(x)$ is denoted by $f(x) = F'(x)$. According to [8], and [34], the probabilistic expressions used to determine the probabilistic models of the PAND and FDEP gates are:

$$Pr\{a \cdot b\}(t) = F_a(t) \times F_b(t), \tag{4.3}$$

$$Pr\{a + b\}(t) = F_a(t) + F_b(t) - F_a(t) \times F_b(t), \tag{4.4}$$

$$Pr\{a \triangleleft b\}(t) = \int_0^t f_a(u)\,(1 - F_b(u))\,du. \tag{4.5}$$

According to [6], and using equations (4.1) and (4.5), the probabilistic model of a PAND gate with independent input events can be given by

$$\begin{aligned} PAND : F_Q(t) = Pr\{Q\}(t) &= Pr\{B \cdot (A \triangleleft B)\}(t) \\ &= \int_0^t f_B(u)\,F_A(u)\,du \\ &= \int_0^t f_B(u)\left(\int_0^u f_A(v)\,dv\right)du. \end{aligned} \tag{4.6}$$

In the case of the FDEP gate, using equations (4.2), (4.3), and (4.4), its probabilistic model can be determined by

$$FDEP : \begin{cases} F_{A_T}(t) = Pr\{A_T\}(t) = Pr\{T + A\}(t) \\ \qquad\qquad = F_A(t) + F_T(t) - F_A(t) \times F_T(t), \\ \\ F_{B_T}(t) = Pr\{B_T\}(t) = Pr\{T + B\}(t) \\ \qquad\qquad = F_B(t) + F_T(t) - F_B(t) \times F_T(t). \end{cases} \tag{4.7}$$

4.3 Algebraic analysis of the DFTA for avionic systems

In this section, the avionic system failure probability of an aircraft is obtained through algebraic analysis of a DFT that is designed using PAND and FDEP dynamic gates along with OR static gates.

Fig. 4.2 shows the illustrates the schematic of the DFT of the aircraft consisting of four subtrees which models failures in different sections of the avionic control systems of the aircraft.

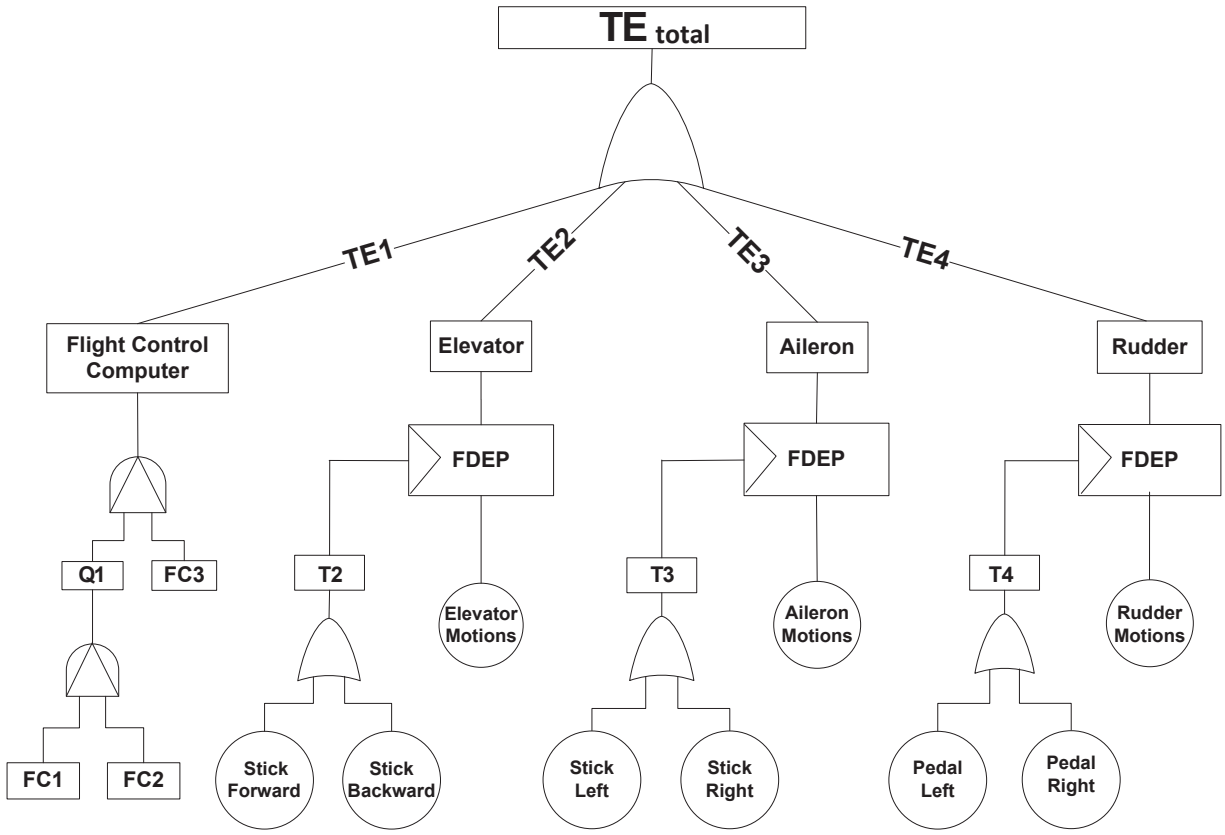


Figure 4.2 Schematic illustration of the DFT for an avionic system.

As shown in Fig. 4.2, the DFT consists of four independent subtrees starting from the left side of the figure:

- Subtree 1: Flight control computers failure with the top event $TE1$,
- Subtree 2: Elevators failure with the top event $TE2$,
- Subtree 3: Ailerons failure with the top event $TE3$,
- Subtree 4: Rudder failure with the top event $TE4$.

The failure probabilities of the four subtrees can be determined by using the probabilistic models of the corresponding gates, i.e., PAND, FDEP and OR gates, given by equations (4.6) and (4.7), respectively. As can be seen in Fig. 4.2, subtree 1 consists of a cascade of two PAND gates to represent the three flight control computers $FC1$, $FC2$, and $FC3$ of the aircraft. During a flight if $FC1$ fails $FC2$ will take over, and in the event that $FC2$ also fails, $FC3$ will be solicited. The failure of $FC3$ after $FC1$ and $FC2$, respectively, results in

subtree 1 failure ($TE1$) which highlights how the order of occurrence in the corresponding dynamic gates leads to failure of subtree 1.

The structure of subtrees 2, 3 and 4 consist of the same gates: a cascade of an OR gate and an FDEP gate with one trigger event and one basic event. The output of the OR gate serves as the trigger of the FDEP gate. Subtrees 2 and 3 model failure probabilities of elevators and ailerons in the aircraft with the top events $TE2$ and $TE3$, respectively. Both elevators and ailerons are activated by their corresponding control stick by means of electronic and hydraulic systems. The elevators are used for controlling the pitch of the aircraft. Applying forward pressure to the control stick, moves the elevators downward, whereas backward pressure on the control stick moves it upward. The ailerons which are moveable plates at the outer trailing edge of the wings are used to control the movement of the aircraft around its longitudinal axis, i.e., roll control. The controlling mechanism of the ailerons is the same as the elevators. Finally, subtree 4 represents the failure probability of the rudder which has the same gate structure as subtrees 2 and 3. The rudder is a movable plate mounted on a fixed surface of the vertical tail unit to manipulate the movement of the aircraft around its vertical axis with the right and left pedals for swinging from side to side, which is referred to as yaw control.

Given the dynamic fault tree of the avionic system illustrated in Fig. 4.2, the event $TE1$ is the top event of two cascaded PAND gates. Using equations (4.1) and (4.6), and considering the fact that the cascaded PAND gates in the DFT have independent input events ($FC1$, $FC2$, and $FC3$) the behavioral model of the event $TE1$ can be expressed as

$$\begin{aligned} TE1 &= FC3 \cdot \Big(Q1 \trianglelefteq FC3 \Big) \\ &= FC3 \cdot \Big((FC2 \cdot (FC1 \trianglelefteq FC2)) \trianglelefteq FC3 \Big) \\ &= FC3 \cdot (FC1 \triangleleft FC2) \cdot (FC2 \triangleleft FC3). \end{aligned} \tag{4.8}$$

where, $Q1$ is the output of the PAND gate with basic events $FC1$, and $FC2$ as shown in Fig. 4.2. According to equations (3.38), (3.39) and (4.6), the failure probability of $TE1$ can be determined by

$$
\begin{aligned}
Pr\{TE1\}(t) &= \int_0^t f_{FC3}(u) \left(\int_0^u f_{FC2}(v)\, F_{FC1}(v)\, dv \right) du \\
&= \int_0^t f_{FC3}(u) \left(\int_0^u f_{FC2}(v) \left(\int_0^v f_{FC1}(w)\, dw \right) dv \right) du \\
&= \int_0^t \left(\int_0^u \left(\int_0^v f_{FC1}(w)\, dw \right) f_{FC2}(v)\, dv \right) f_{FC3}(u)\, du.
\end{aligned}
\tag{4.9}
$$

which can be expressed as

$$
\begin{aligned}
Pr\{TE1\}(t) = {} & \frac{\lambda_{FC1}\lambda_{FC2}}{\lambda_{FC2}+\lambda_{FC3}} \int_0^t e^{-u(\lambda_{FC1}+\lambda_{FC2}+\lambda_{FC3})}\, du - \lambda_{FC1} \int_0^t e^{-u(\lambda_{FC3}\, t+(\lambda_{FC1}+\lambda_{FC2}))}\, du \\
& + \frac{\lambda_{FC1}\lambda_{FC3}}{\lambda_{FC2}+\lambda_{FC3}} \int_0^t e^{-((\lambda_{FC2}+\lambda_{FC3})\, t+\lambda_{FC1}\, u)}\, du \\
= {} & \frac{\lambda_{FC1}\lambda_{FC2}}{(\lambda_{FC2}+\lambda_{FC3})(\lambda_{FC1}+\lambda_{FC2}+\lambda_{FC3})}(1 - e^{-(\lambda_{FC1}+\lambda_{FC2}+\lambda_{FC3})\, t}) \\
& - \frac{\lambda_{FC1}(e^{-\lambda_{FC3}\, t} - e^{-(\lambda_{FC1}+\lambda_{FC2}+\lambda_{FC3})\, t})}{(\lambda_{FC1}+\lambda_{FC2})} \\
& + \frac{\lambda_{FC3}(e^{-(\lambda_{FC2}+\lambda_{FC3})\, t} - e^{-(\lambda_{FC1}+\lambda_{FC2}+\lambda_{FC3})\, t})}{(\lambda_{FC2}+\lambda_{FC3})} \\
= {} & \frac{\lambda_{FC1}\lambda_{FC2}}{(\lambda_{FC2}+\lambda_{FC3})(\lambda_{FC1}+\lambda_{FC2}+\lambda_{FC3})} \\
& - \frac{\lambda_{FC1}}{(\lambda_{FC1}+\lambda_{FC2})} e^{-\lambda_{FC3}\, t} + \frac{\lambda_{FC3}}{(\lambda_{FC2}+\lambda_{FC3})} e^{-(\lambda_{FC2}+\lambda_{FC3})\, t} \\
& - \frac{\lambda_{FC2}\lambda_{FC3}\, e^{-(\lambda_{FC1}+\lambda_{FC2}+\lambda_{FC3})\, t}}{(\lambda_{FC1}+\lambda_{FC3})(\lambda_{FC1}+\lambda_{FC2}+\lambda_{FC3})}.
\end{aligned}
\tag{4.10}
$$

where λ_{FC1}, λ_{FC2}, and λ_{FC3} represent the failure rates of $FC1$, $FC2$, and $FC3$, respectively. Fig. 4.3 shows the probability of failure trends when branch $TE1$ is affected by different failure rates of its corresponding basic events.

As expected when the failure rates of the basic events ($\lambda = \lambda_{FC1} = \lambda_{FC2} = \lambda_{FC3}$) increase, the probability of failure reaches one more rapidly.

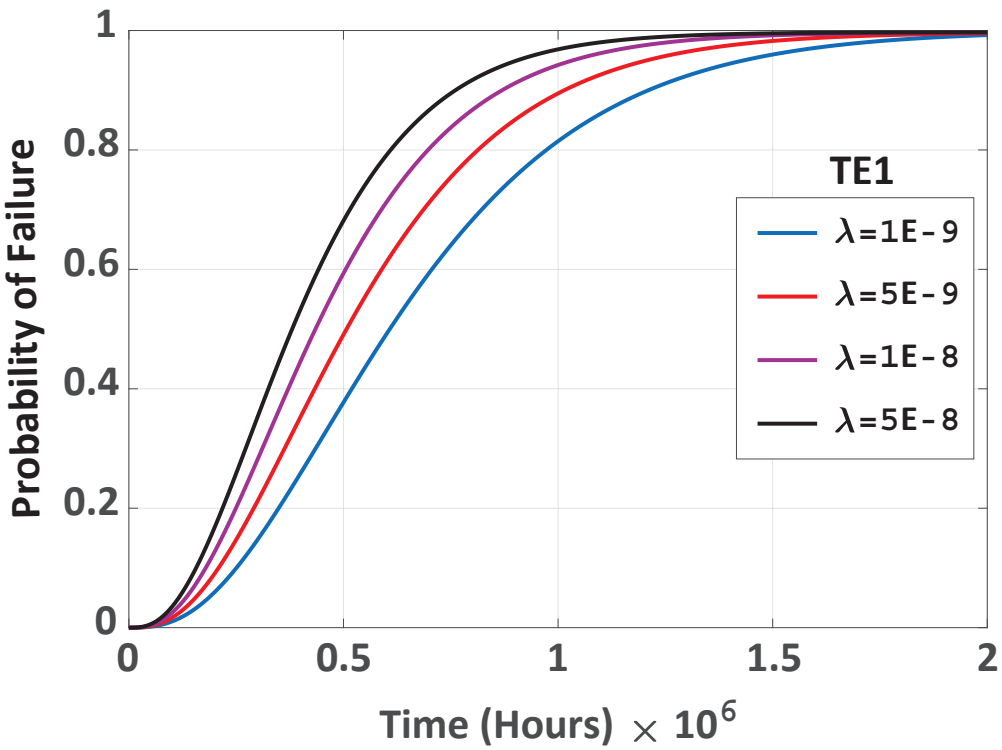


Figure 4.3 Probability of failure of $TE1$ as failure rates ($\lambda = \lambda_{FC1} = \lambda_{FC2} = \lambda_{FC3}$) increases.

In the case of the events $TE2$, $TE3$, and $TE4$, the corresponding subtrees are composed of an OR gate where the output serves as a trigger event of an FDEP gate. In each branch, the FDEP gate has one dependent basic event, i.e., Elevator Motions, Aileron Motions or Rudder Motions, being trigged by the trigger event $T2$, $T3$ and $T4$, respectively. The dependent basic event takes place once the corresponding trigger event occurs. Therefore, using equations (4.2), (4.4) and (4.7), the probability of failure of $TE2$ can be given by

$$\begin{aligned} Pr\{TE2\}(t) &= Pr\{E + T2\}(t) \\ &= 1 - \Big(1 - Pr\{E(t)\}\Big)\Big(1 - Pr\{T2(t)\}\Big) \\ &= 1 - \Big(1 - F_E(t)\Big)\Big(1 - F_{T2}(t)\Big) \\ &= F_E(t) + F_{T2}(t) - F_E(t) \times F_{T2}(t) \\ &= F_E(t) + F_{SB}(t) + F_{SF}(t) \\ &\quad - F_E(t) \times F_{SB}(t) - F_E(t) \times F_{SF}(t) - F_{SB}(t) \times F_{SF}(t) \\ &\quad + F_E(t) \times F_{SB}(t) \times F_{SF}(t). \end{aligned} \tag{4.11}$$

where E, SB, and SF represent elevator motions, stick backward and stick forward events in $TE2$, respectively. As shown in Fig. 4.2, the output of the OR gate ($T2$) serves as the trigger of the dependent event E in the FDEP gate. The dependent basic event E occurs once its trigger event $T2$ takes place.

Figs. 4.4 and 4.5 show the probability of failure behavior of $TE2$ as the failure rate of its basic events and the dependent event changes, respectively. According to Fig. 4.4, when the failure rate for the basic events of the OR gate (SB and SF) is higher than that of the dependent event of the FDEP gate (E), the failure probability of the subteree asymptotically converges to one faster.

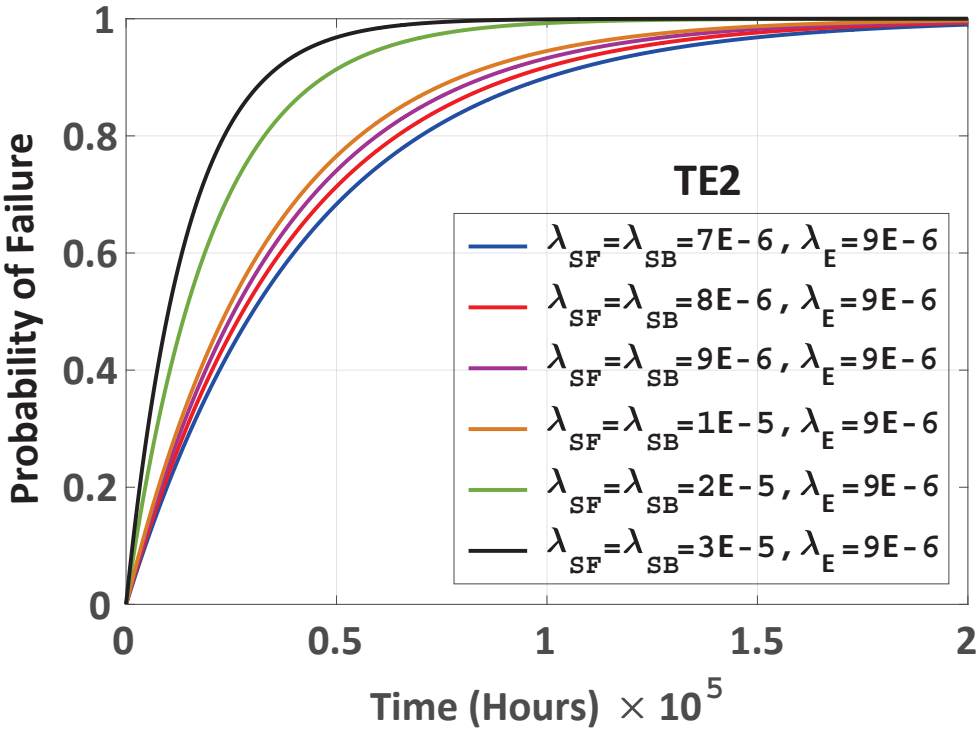


Figure 4.4 Failure probabilities of $TE2$ for different $\lambda_{SF} = \lambda_{SB}$,i.e., failure rates of its basic events SF and SB of the OR gate of gate FDEP.

As shown in Fig. 4.5, similar scenario takes place in the case of higher failure rates for E compared to that of SF and SB but with a lower acceleration.

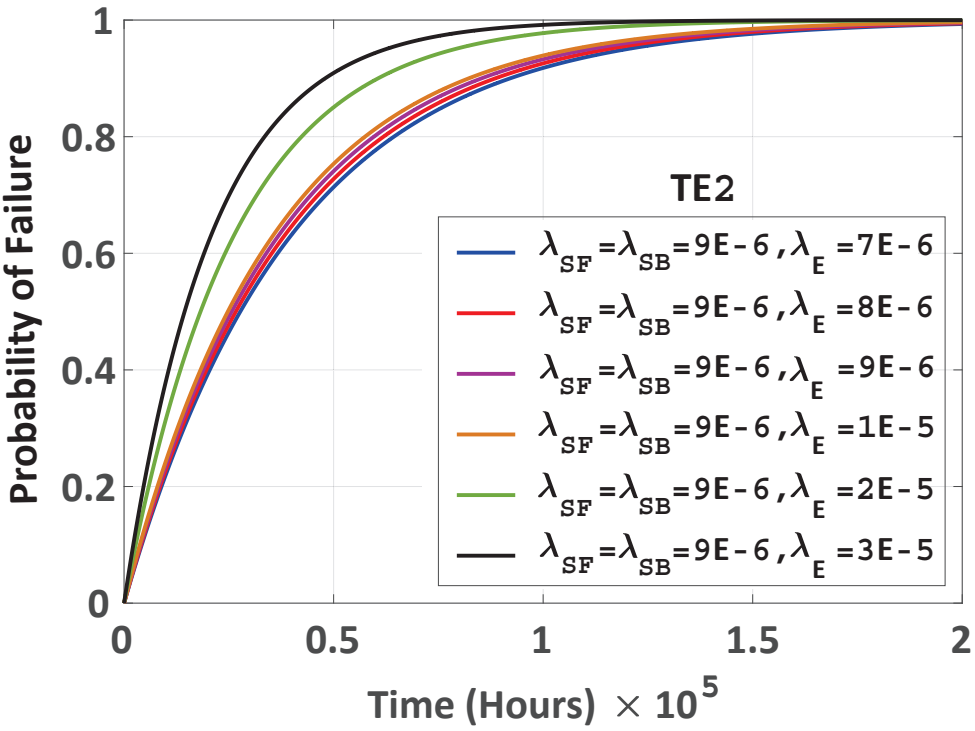


Figure 4.5 Failure probabilities of $TE2$ for different λ_E ,i.e., failure rates of the dependent event E of gate FDEP.

Similarly, the failure probability of $TE3$ and $TE4$ can be obtained by

$$\begin{aligned} Pr\{TE3\}(t) &= Pr\{Ai + T3\}(t) \\ &= F_{Ai}(t) + F_{SR}(t) + F_{SL}(t) \\ &-F_{Ai}(t) \times F_{SR}(t) - F_{Ai}(t) \times F_{SL}(t) - F_{SR}(t) \times F_{SL}(t) \\ &+F_{Ai}(t) \times F_{SR}(t) \times F_{SL}(t), \end{aligned} \tag{4.12}$$

and

$$\begin{aligned} Pr\{TE4\}(t) &= Pr\{R + T4\}(t) \\ &= F_{R}(t) + F_{PR}(t) + F_{PL}(t) \\ &-F_{R}(t) \times F_{SR}(t) - F_{R}(t) \times F_{SL}(t) - F_{SR}(t) \times F_{SL}(t) \\ &+F_{R}(t) \times F_{SR}(t) \times F_{SL}(t), \end{aligned} \tag{4.13}$$

where Ai, SR, SL denote aileron motions, stick right and stick left events in $TE3$, and R, PR, PL represent rudder motions, pedal right and pedal left events in $TE4$, respectively. Therefore, using the inclusion-exclusion equation described in [6, 35], the entire failure probability of the DFT shown in Fig. 4.2 with the top event, TE_{Total}, can be expressed as

$$
\begin{aligned}
&Pr\{TE_{total}\} = Pr\{TE1 + TE2 + TE3 + TE4\}(t) \\
&= 1 - \Big[\Big(1 - Pr\{TE1\}(t)\Big)\Big(1 - Pr\{TE2\}(t)\Big)\Big(1 - Pr\{TE3\}(t)\Big)\Big(1 - Pr\{TE4\}(t)\Big)\Big] \\
&= Pr\{TE1\}(t) + Pr\{TE2\}(t) + Pr\{TE3\}(t) + Pr\{TE4\}(t) \\
&- Pr\{TE1\}(t) \times Pr\{TE2\}(t) - Pr\{TE1\}(t) \times Pr\{TE3\}(t) \\
&- Pr\{TE1\}(t) \times Pr\{TE4\}(t) - Pr\{TE2\}(t) \times Pr\{TE3\}(t) \\
&- Pr\{TE2\}(t) \times Pr\{TE4\}(t) - Pr\{TE3\}(t) \times Pr\{TE4\}(t) \\
&+ Pr\{TE1\}(t) \times Pr\{TE2\}(t) \times Pr\{TE3\}(t) \\
&+ Pr\{TE1\}(t) \times Pr\{TE2\}(t) \times Pr\{TE4\}(t) \\
&+ Pr\{TE1\}(t) \times Pr\{TE3\}(t) \times Pr\{TE4\}(t) \\
&+ Pr\{TE2\}(t) \times Pr\{TE3\}(t) \times Pr\{TE4\}(t) \\
&- Pr\{TE1\}(t) Pr\{TE2\}(t) \times Pr\{TE3\}(t) \times Pr\{TE4\}(t).
\end{aligned}
\tag{4.14}
$$

In this thesis, to determine the failure probability of the DFT for the avionic system, we make use of the standard failure rates (failures/hour) of the constituent components given in [1]. Table 4.1 summarizes the standard failure rates for the different components in the DFT of the avionic system given in Fig. 4.2.

Table 4.1 Standard failure rates (failures$\times$hour^{-1}) of the basic events in the DFT from [1].

	Flight Control Computer			**Elevator**			**Aileron**			**Rudder**		
Components	$FC1$	$FC2$	$FC3$	SF	SB	E	SL	SR	Ai	PL	PR	R
λ (failures$\times$hour^{-1})	1E-9	1E-9	1E-9	8E-6	8E-6	1E-7	9E-6	9E-6	1E-7	1E-5	1E-5	1E-7

Using these standard failure rates, failure probabilities of the four subtrees in the DFT, shown in Fig. 4.2, can be individually determined as a function of time. These obtained values can then be used in equation (4.14) to obtain the failure probability of the whole system $Pr\{TE_{Total}\}(t)$.

Fig. 4.6 shows the probability of failure obtained for the four subtrees and for the whole avionic system as a function of time in hours. It can be inferred from the figure that subtree 2 to 4 follow the same trend as they have the same architecture. The only difference comes from the speed at which they converge towards one in accordance to the difference in stick and pedal failure rates employed in the model as displayed in Table 4.1.

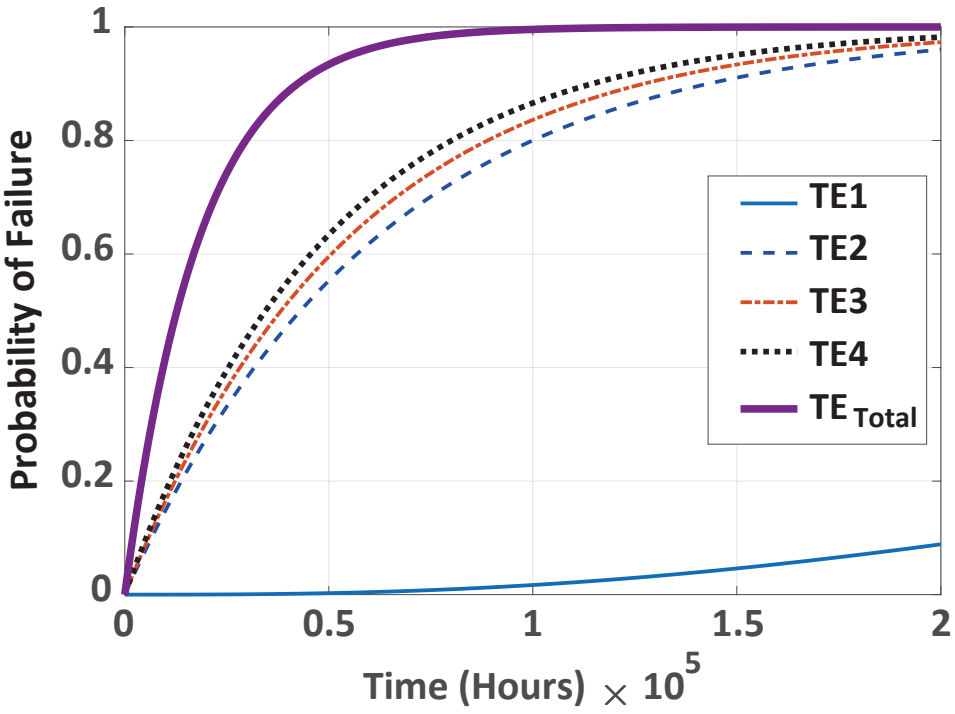


Figure 4.6 Failure probabilities of the four subtrees separately, and the TE.

In the case of subtree $TE1$, given the triple redundancy dedicated to the flight control computers, the failure rate of the flight control system increases with time much slower compared to the other branches of the system. Fig. 4.7 shows the probability of failure for $TE1$ on a longer time scale, proving that it will eventually reach one as expected for every branch of the system.

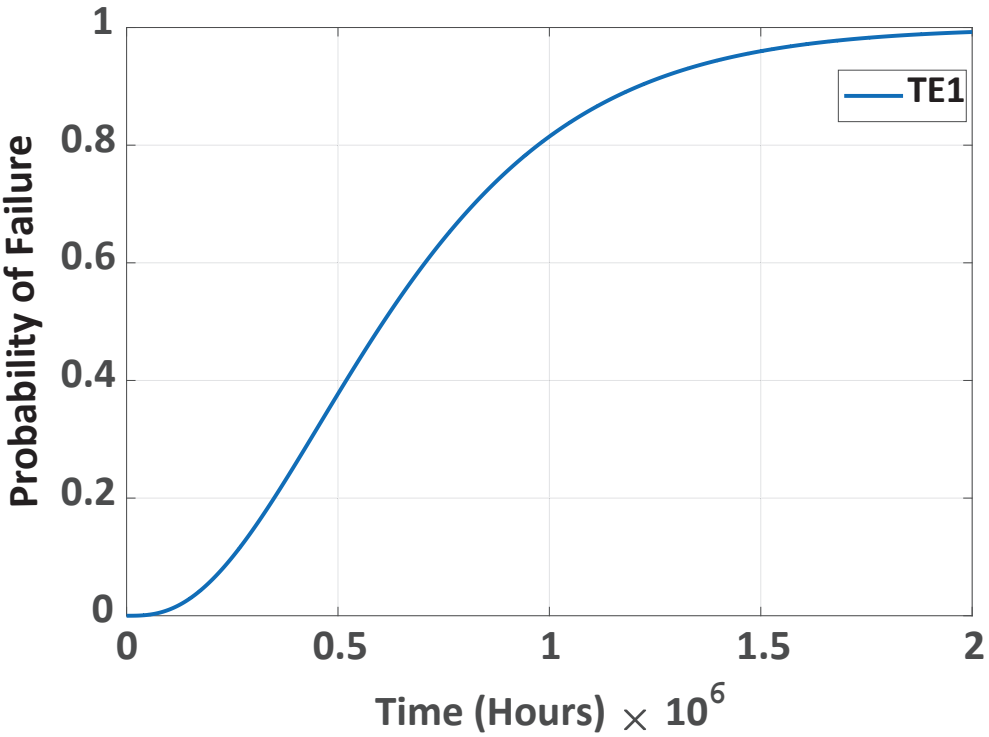


Figure 4.7 Failure probabilities of $TE1$ which is on a longer time scale compared to that of shown in Fig. reffig:FailureProbabilities.

The presented DFTA can be extended to a system with many independent subtrees to model failure in avionic systems, which allows for predicting failure probability of different software and hardware components in aircrafts. Depending on the avionic system components to be modeled and included in the DFT, different structure functions can be developed based on the dynamic and static gates used in the related subtrees. In the DFTA, the results of independent subtrees can be simply integrated in equation (4.14) to determine the failure probability, which reduces the complexity of the system analysis. Using this technique, failure mechanisms in different avionic systems, such as avionic display system, electronic flight control systems, flight guidance system, electric power supply system etc., can be modeled and evaluated to minimize the risks of any malfunction in a specific component.

Furthermore, some errors or failures may exist in requirements, design, function and test of these systems will be the fundamental cause of accident. The mistakes arising from design and development process, should be detected and addressed before the system is in operation. However, there might be some undetected errors or some perfectly functioning components which are designed, manufactured and tested earlier may still contain logic errors. Regardless of the source of errors, they may propagate to other sections in the aircraft during a flight.

Therefore, the proposed DFTA would be a promising method to define all types of errors, faults, and failures in avionic system components that may place the aircraft in a hazardous condition (TE).

4.4 Conclusion

A DFT to model failures in avionic systems was designed and investigated. The failure probability of the DFT structure was determined by an algebraic analysis including four subtrees of the avionic system: flight control, aileron, rudder, and elevator. To this end, temporal operators such as non-inclusive BEFORE ($\lhd$), SIMULTANOUS ($\triangle$) and Inclusive BEFORE ($\trianglelefteq$) along with some other algebraic theorems were used to define the behavioral and probabilistic models of the employed dynamic gates, i.e., PAND and FDEP gates, as the building-blocks of the DFT. The results show the consistency of the model with respect to the failure rates and the level of the redundancy defined for the subtrees.

CHAPTER 5 CONCLUSION

This thesis presents a comprehensive methodology to propose a new design of DFT for avionic systems. The DFT investigates the probability of failure due to a fault in the motion control systems and the flight control computers of the aircraft. An aircraft has several circuit components which play key roles in steering and stabilizing it, such as Elevators, Rudder, and Ailerons to perform pitch control, roll control and yaw control, respectively. Additionally, there is a redundancy of three flight control computers in the aircraft. Initially, the failure rate of these components should be well investigated to ensure accurate modeling. In this regard, the failure of each component is considered as basic events of specific dynamic gates and subtrees to algebraically determine the behavioral model of the subsystem.
The failure in each flight control component is separately modeled by employing the appropriate dynamic gates. Using the behavioral model and probabilistic models of each dynamic gate, the system reliability is calculated algebraically on the basis of the related temporal operators and mathematical theorems. As a result, the precise failure probability of the corresponding subtree in the DFT can be determined. Using the obtained expressions of the behavioral models and failure probabilities of the subtrees, the total failure of the avionic systems in the aircraft can be expressed as an algebraic expression for the top event of the DFT.

The results of this research work show the consistency of the model with respect to the failure rates and the level of the redundancy defined for the subtrees. Being less computer intensive, the proposed algebraic method is able to solve the DFT of the avionic system fast and accurately. The designed DFT for an aircraft and its algebraic analysis allows for obtaining the total failure probability of the plane which is vital to minimize the risk of fatal air crashes due to defects in different flight components. As a result, the failures can be anticipated before happening and the appropriate solutions can be considered to enhance the safety of the aircraft to a certain extent.

5.1 Publications

This research work was presented at *The Canadian Aeronautics and Space Institute's (CASI) AERO 2019*, and has been published in the proceeding.

5.2 Future Research

As a future work, the system can be designed to predict the probability of failure for various avionic system components such as flight guidance system, electric power supply system, on-board radiation, etc.

REFERENCES

[1] V. Hilderman, "Do-178b to do-178c: Impact on avionics verification & certification," 2011.

[2] M. Hamaidia, M. Kara, and F. Innal, "Probability and frequency derivation using dynamic fault frees," *Process Safety Progress*, 2018.

[3] S. Kabir *et al.*, "Uncertainty-aware dynamic reliability analysis framework for complex systems," *IEEE Access*, 2018.

[4] A. Dutle, C. Muñoz, and A. Narkawicz, *NASA Formal Methods: 10th International Symposium, NFM 2018, Newport News, VA, USA, April 17-19, 2018, Proceedings*. Springer, 2018, vol. 10811.

[5] J. B. Dugan, S. J. Bavuso, and M. A. Boyd, "Dynamic fault-tree models for fault-tolerant computer systems," *IEEE Transactions on reliability*, vol. 41, no. 3, pp. 363–377, 1992.

[6] G. Merle *et al.*, "Probabilistic algebraic analysis of fault trees with priority dynamic gates and repeated events," *IEEE Transactions on Reliability*, vol. 59, no. 1, pp. 250–261, 2010.

[7] G. Merle, "Algebraic modelling of dynamic fault trees, contribution to qualitative and quantitative analysis," Ph.D. dissertation, École normale supérieure de Cachan-ENS Cachan, 2010.

[8] S. Amari, G. Dill, and E. Howald, "A new approach to solve dynamic fault trees," in *Reliability and Maintainability Symposium, 2003. Annual.* IEEE, 2003, pp. 374–379.

[9] M. Stamatelatos *et al.*, "Fault tree handbook with aerospace applications," 2002.

[10] M. Walker and Y. Papadopoulos, "Qualitative temporal analysis: Towards a full implementation of the fault tree handbook," *Control Engineering Practice*, vol. 17, no. 10, pp. 1115–1125, 2009.

[11] T. Yuge and S. Yanagi, "Quantitative analysis of a fault tree with priority and gates," *Reliability Engineering & System Safety*, vol. 93, no. 11, pp. 1577–1583, 2008.

[12] K. D. Rao *et al.*, "Dynamic fault tree analysis using monte carlo simulation in probabilistic safety assessment," *Reliability Engineering & System Safety*, vol. 94, no. 4, pp. 872–883, 2009.

[13] L. Meshkat, *Dependency modeling and phase analysis for embedded computer based systems*. University of Virginia, 2000.

[14] M. Čepin and B. Mavko, "A dynamic fault tree," *Reliability Engineering & System Safety*, vol. 75, no. 1, pp. 83–91, 2002.

[15] R. Manian *et al.*, "Bridging the gap between systems and dynamic fault tree models," in *Reliability and Maintainability Symposium, 1999. Proceedings. Annual.* IEEE, 1999, pp. 105–111.

[16] M. Marseguerra *et al.*, "A concept paper on dynamic reliability via monte carlo simulation," *Mathematics and Computers in Simulation*, vol. 47, no. 2, pp. 371–382, 1998.

[17] L. Meshkat, J. B. Dugan, and J. D. Andrews, "Dependability analysis of systems with on-demand and active failure modes, using dynamic fault trees," *IEEE Transactions on Reliability*, vol. 51, no. 2, pp. 240–251, 2002.

[18] J. B. Dugan, K. J. Sullivan, and D. Coppit, "Developing a low-cost high-quality software tool for dynamic fault-tree analysis," *IEEE Transactions on reliability*, vol. 49, no. 1, pp. 49–59, 2000.

[19] C.-Y. Huang and Y.-R. Chang, "An improved decomposition scheme for assessing the reliability of embedded systems by using dynamic fault trees," *Reliability Engineering & System Safety*, vol. 92, no. 10, pp. 1403–1412, 2007.

[20] A. Bobbio and D. C. Raiteri, "Parametric fault trees with dynamic gates and repair boxes," in *Reliability and Maintainability, 2004 Annual Symposium-RAMS.* IEEE, 2004, pp. 459–465.

[21] A. Bobbio *et al.*, "Improving the analysis of dependable systems by mapping fault trees into bayesian networks," *Reliability Engineering & System Safety*, vol. 71, no. 3, pp. 249–260, 2001.

[22] L. Meshkat *et al.*, "An overview of the phase-modular fault tree approach to phased mission system analysis," 2003.

[23] Y. Dutuit and A. Rauzy, "A linear-time algorithm to find modules of fault trees," *IEEE Transactions on Reliability*, vol. 45, no. 3, pp. 422–425, 1996.

[24] R. Gulati and J. B. Dugan, "A modular approach for analyzing static and dynamic fault trees," in *Reliability and Maintainability Symposium. 1997 Proceedings, Annual.* IEEE, 1997, pp. 57–63.

[25] A. C. Marquez, A. S. Heguedas, and B. Iung, "Monte carlo-based assessment of system availability. a case study for cogeneration plants," *Reliability Engineering & System Safety*, vol. 88, no. 3, pp. 273–289, 2005.

[26] E. Zio, L. Podofillini, and V. Zille, "A combination of monte carlo simulation and cellular automata for computing the availability of complex network systems," *Reliability Engineering & System Safety*, vol. 91, no. 2, pp. 181–190, 2006.

[27] G. Merle, J.-M. Roussel, and J.-J. Lesage, "Improving the efficiency of dynamic fault tree analysis by considering gate fdep as static," in *European Safety and Reliability Conference (ESREL 2010)*. Taylor & Francis, 2010, pp. pp–845.

[28] H. Aliee and H. R. Zarandi, "A fast and accurate fault tree analysis based on stochastic logic implemented on field-programmable gate arrays," *IEEE Transactions on Reliability*, vol. 62, no. 1, pp. 13–22, 2013.

[29] A. Ejlali and S. G. Miremadi, "Fpga-based monte carlo simulation for fault tree analysis," *Microelectronics Reliability*, vol. 44, no. 6, pp. 1017–1028, 2004.

[30] H. Boudali, P. Crouzen, and M. Stoelinga, "Dynamic fault tree analysis using input/output interactive markov chains," in *Dependable Systems and Networks, 2007. DSN'07. 37th Annual IEEE/IFIP International Conference on*. IEEE, 2007, pp. 708–717.

[31] L. Xing and G. Levitin, "Combinatorial algorithm for reliability analysis of multistate systems with propagated failures and failure isolation effect," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 6, pp. 1156–1165, 2011.

[32] G. Merle and J.-M. Roussel, "Algebraic modelling of fault trees with priority and gates," in *1st IFAC Workshop on Dependable Control of Discrete Systems (DCDS'07)*, 2007, pp. pp–175.

[33] G. Merle *et al.*, "Algebraic expression of the structure function of a subclass of dynamic fault trees," in *2nd IFAC Workshop on Dependable Control of Discrete Systems (DCDS'09)*, 2009, pp. 129–134.

[34] J. Fussell, E. Aber, and R. Rahl, "On the quantitative analysis of priority-and failure logic," *IEEE Transactions on Reliability*, vol. 25, no. 5, pp. 324–326, 1976.

[35] K. S. Trivedi, *Probability & statistics with reliability, queuing and computer science applications*. John Wiley & Sons, 2008.