Remenyte-Prescott, Rasa and Andrews, John and Chung, Paul (2010) An efficient phased mission reliability analysis for autonomous vehicles. Reliability Engineering and System Safety, 95 (3). pp. 226-235. ISSN 0951-8320

An Efficient Phased Mission Reliability Analysis for Autonomous Vehicles

R.Remenyte-Prescott[1*], J.D.Andrews[1], P.W.H.Chung[2]

*corresponding author:
Tel: +441158467366
Fax: +441159513909
Email: R.Remenyte-Prescott@nottingham.ac.uk

[1]Nottingham Transportation Engineering Centre, Faculty of Engineering, University of Nottingham, Nottingham, NG7 2RD, England, UK

[2]Dep. of Computer Science, Loughborough University, Loughborough, LE11 3TU, England, UK

Abstract

Autonomous systems are becoming more commonly used, especially in hazardous situations. Such systems are expected to make their own decisions about future actions when some capabilities degrade due to failures of their subsystems. Such decisions are made without human input, therefore they need to be well-informed in a short time when the situation is analysed and future consequences of the failure are estimated. The future planning of the mission should take account of the likelihood of mission failure. The reliability analysis for autonomous systems can be performed using the methodologies developed for phased mission analysis, where the causes of failure for each phase in the mission can be expressed by fault trees.

Unmanned Autonomous Vehicles (UAVs) are of a particular interest in the aeronautical industry, where it is a long term ambition to operate them routinely in civil airspace. Safety is the main requirement for the UAV operation and the calculation of failure probability of each

phase and the overall mission is the topic of this paper. When components or sub-systems fail or environmental conditions throughout the mission change, these changes can affect the future mission. The new proposed methodology takes into account the available diagnostics data and is used to predict future capabilities of the UAV in real-time. Since this methodology is based on the efficient BDD method, the quickly provided advice can be used in making decisions. When failures occur appropriate actions are required in order to preserve safety of the autonomous vehicle. The overall decision making strategy for autonomous vehicles is explained in this paper. Some limitations of the methodology are discussed and further improvements are presented based on experimental results.

Keywords: phased mission, autonomous system, fault tree, binary decision diagram, reliability

1. Introduction

A phased mission describes a situation when the requirements for success change throughout the time of operation, therefore, the causes of failure during the mission also change. The consecutive and distinct periods in the mission performing different tasks are known as phases, performed in sequence. In order for the mission to be successful, each of the phases must be completed successfully, therefore, the mission fails if at least one phase fails. Many systems operate in this way, with a typical example being an aircraft flight, containing a number of phases, such as: taxi, take-off, climb to required altitude, cruise, descend, landing and taxi back to the terminal. The mission can fail in any of the phases and the purpose of the analysis is to predict phase failure probabilities, which are added to obtain the overall mission failure probability.

Autonomous systems are becoming increasingly useful in today's society because they can operate in conditions which would be risky for humans. For such autonomous systems the requirement is placed on the system to make its own decisions, without human help, at different stages of the mission. These decisions about immediate or future actions must be well-informed, considering the risk related to the platform itself and objects located close to the location of the platform. The rational decisions are expected to preserve the safety of the operation and achieve the mission success.

A significant factor in the decision making process of autonomous systems is the mission failure probability during the future phases. There are two types of predictions required – before the start of the mission and while the mission is in progress. The initial mission failure probability, that determines if the mission should start in the current system configuration, changes throughout the course of the mission. The updates are calculated when certain phases have been completed successfully and some components that affect the requirements of phase success have failed. If the updated failure probability is unacceptably high, the mission cannot proceed in its current configuration and alternatives should be used. An example of such an alternative for a UAV could be landing in a different airport than that initially intended. One of the requirements for this strategy is to be able to adapt rapidly to changing mission environment and diagnostics data, which report the status of components, functions or subsystems. Therefore, a prognostics tool is expected to provide accurate information in a short time so that the decision making process would be well-informed and appropriate decisions would be made before a catastrophic event.

Previously developed risk assessment methods are used in the phased mission analysis. Fault tree analysis is suitable when describing non-repairable systems in [1], [2] and [3], where component failures are treated independently. Fault trees code the failure logic in a well-documented way, however, their limitations become apparent when performing the

quantitative analysis. Binary Decision Diagrams (BDDs) were developed as an alternative logic function representation in [4], [5] and [6]. Due to the efficiency of the analysis method for BDDs they have also been applied to phased mission analysis, such as [7] and [8]. Also, prior to fault tree conversion to BDD fault tree simplifications can be applied, as shown in [9] and [10], which result in a more concise fault tree form. These techniques can be also applied to phase fault trees, where independent modules are identified, resulting in smaller fault trees and BDDs. This approach is applied in the proposed methodology.

A mission planning strategy is presented in this paper, which is based on using the prognostics methodology for autonomous vehicles (primarily UAVs) performing phased missions. The ability to calculate the initial mission failure probability [11] is now enhanced to calculate the updated mission failure probability, when new information about the health of the system and operational conditions is obtained. This contribution is particularly important if the phased mission modelling is to be applied as a prognostics tool in the mission planning of autonomous vehicles. In such application the mission failure probability is revaluated throughout the mission according to fault diagnostics information. The novelty of this method is its simplicity when connecting phase failure BDDs and its ability to take into account not only system components (single/multiple failure mode) but also external factors. In addition, this paper shows how to calculate the updated future phase failure probabilities after each phase is completed successfully. Such analysis is possible, since not only the overall mission failure probability but also each phase failure probabilities are calculated using this method. Also, the increased efficiency of the BDD technique for predictions of mission failure probability is achieved, when phase fault trees are simplified prior to the conversion to BDDs. A fast and efficient phased mission reliability analysis can be used to support the decision making process of autonomous vehicles.

2. Phased Mission Modelling

Phased missions are used to define the behaviour of a system during different parts of the mission and to perform the analysis. The following characteristics determine a phased mission:

- Every mission consists of many consecutive phases performed in sequence.
- Since a different task is to be performed in each phase there are different failure criteria in each phase.
- For a mission to be successful all phases must be completed successfully.

Further assumptions are made in this paper. First of all, the length of each phase is known. Secondly, before the start of the mission all components are considered to be working. And finally, since the non-repairable mission is considered component failures remain present in the system after they have happened.

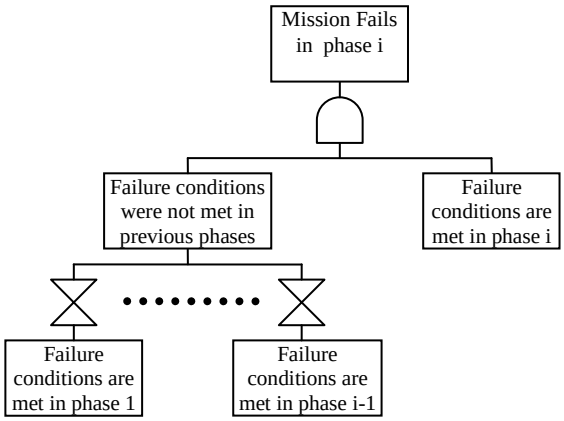

Figure 1 – Fault tree for mission failure in phase i, Ph_i

The phased mission is represented by a number of fault trees, each of them expressing the conditions leading to the failure of a phase. For any phase the method, proposed in [8], combines the causes of success of previous phases with the causes of failure for the phase being considered. A general fault is shown in Figure 1. As it can be seen from the diagram,

the mission failure logic in phase i is expressed as a conjunction of failure conditions NOT being met in any of the previous phases i – 1 and the failure conditions met in phase i. Let F_i express the logical expression for the failure conditions being met in phase i and Ph_i express the logical expression for mission failure in phase i. Then:.

$$\begin{aligned} Ph_1 &= F_1 \\ &\vdots \\ Ph_i &= \overline{F_1} \cdot \overline{F_2} \cdot \overline{F_3} \cdot \ldots \cdot \overline{F_{i-1}} \cdot F_i \end{aligned} \tag{1}$$

Fault tree analysis can be used to quantify the probability of mission failure during mission phase i, Q_i:

$$Q_i = P(Ph_i). \tag{2}$$

Since the mission fails if at least one of its phases fail, the logical expression for mission failure is given by Ph_{MISS}:

$$Ph_{MISS} = Ph_1 + Ph_2 + ... + Ph_3 \tag{3}$$

The total mission failure probability, Q_{MISS}, is obtained by adding these mission phase failure probabilities, as shown in equation (4):

$$Q_{MISS} = \sum_{i=1}^{n} Q_i \ , \tag{4}$$

where n is the total number of phases in the mission. Since the conditional phase failures are mutually exclusive events the total mission failure probability is simply the sum of mission phase failure probabilities.

Once the mission is underway, Q_i is updated taking into account the success of the previous phases. Using Bayes' theorem gives the expression to calculate the updated phase failure probability, $Q_{j|\bar{k}}$, the probability of failure in phase j given that phase k was successfully completed, as shown in [12]:

$$Q_{j|\bar{k}} = \frac{Q_j}{1 - \sum_{i=1}^{k} Q_i}. \tag{5}$$

Then the overall mission failure probability is calculated by adding the phase failure probabilities of the future phases.

$$Q_{MISS|\bar{k}} = \sum_{j=k+1}^{n} Q_{j|\bar{k}}. \quad (6)$$

This concludes the basic phased mission modelling which is going to be extended for autonomous vehicles.

3. Mission Planning for Autonomous Vehicles

A decision making strategy which can be used in mission planning for autonomous vehicles performing phased missions is presented in this section. This approach is based on the probability of mission failure, which is calculated before the mission starts, and then it is updated throughout the course of the mission. These are explained below together with the requirements for the implementation of the strategy.

3.1. Types of mission failure probability

There are two different probabilities that are required to be calculated during the course of the mission. These are the initial mission failure probability and the updated mission failure probability. The initial mission failure probability is calculated once, before the start of the mission. It gives the likelihood of the mission failure if the current system and mission configuration is used. The updated failure probability can be calculated many times during the mission, for example, after each successfully completed phase. Also as failures occur, as indicated for example by a fault detection system, the updated failure probability gives the indication of how likely is the future mission to failure, when certain failures occur. It is also appropriate if environmental conditions change during the mission.

The mission failure probability is the key factor in the mission planning process, presented in the following section. The two types of probability are calculated and used while making decisions.

3.2. Decision making strategy

The strategy in [13] involves calculating the probability of mission failure at required points during the mission, usually when faults occur in the system or environmental conditions change. If the probability of mission failure becomes too high, then the future mission is considered to be too risky and an alternative mission configuration is used. The strategy is shown in Figure 2.

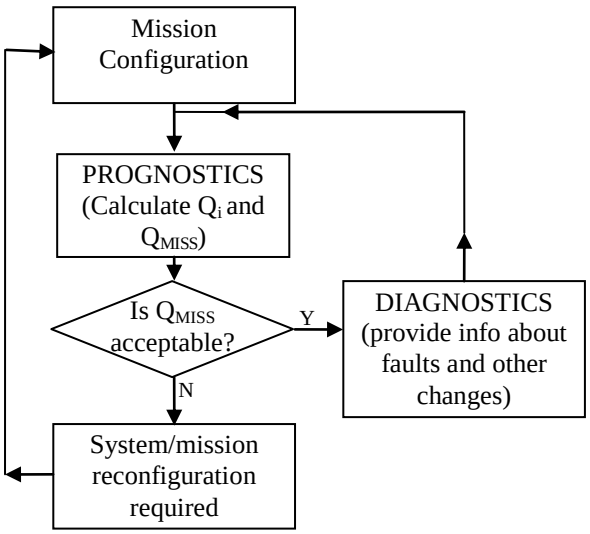

Figure 2 – Decision Making Strategy

There are two main parts in this approach – diagnostics and prognostics. The objectives of the two methods and their interactions are explained later. The prognostics is the focus of this paper. During the diagnostics process the information about the status of the system is collected. The changes can be recorded as faults that occur on the system at component or subsystem level. Changes in environment conditions that affect the operation of the system,

such as weather conditions or other approaching vehicles are also recorded. An important part of this information is successfully completed phases or tasks, confirming components or subsystems functioning successfully up to a certain time in the mission. During the prognostics process all the available information is used to calculate the probability of mission failure. A system reliability technique is employed in this application, when failure probabilities are calculated for each phase in the mission and for the overall mission. The initial and updated probabilities are calculated in this part of the decision making process.

The acceptable mission failure probability is defined before the predictions are obtained. If the initial or updated probabilities exceed the acceptable level of failure, the mission cannot be carried out in its current configuration. It could mean that the system still performs its mission task but in a different way in order to achieve the objective of the mission. Alternatively, the system can be reconfigured to achieve an alternative mission objective or the mission can be aborted and no further actions are performed. The scope of this work is to assume that mission configuration options are known and the most suitable one is chosen according to its failure probability. Therefore, if the decision of the autonomous vehicle is based on this information only, the mission is chosen which is the least likely to fail. However, in other applications, for example, military environment, different criteria might be more important. For example, UAV needs to hit a target in a set time, but is not required to land safely.

In summary, for a defined mission the prognosis would be carried out before the start of the mission, calculating the initial failure probabilities of the original mission configuration. If the mission failure probability is acceptable, then the mission begins. During the mission the diagnostics tool collects the information about all the changes. Using the information available the prognostics tool calculates the updated failure probabilities and again checks

them against the acceptable limits. A mission reconfiguration takes place if the failure probability becomes too high.

3.3. Requirements

It is important to obtain the predictions quickly, especially when the decision making process relies on the speed of the prognosis. The ability to respond rapidly to changes in the status of the system and in the environmental conditions is an important requirement for the prognostics process. If autonomous vehicles are used, their decisions should be well-informed and made in a short time relying on the accurate information. This leads to two main requirements for the reliability-based prognosis – accuracy and speed.

When considering the reliability techniques available, fault trees provide an excellent way to represent the failure logic of each phase. However, when fault trees are used for the analysis, the kinetic tree theory [14] is not suitable to deliver the results required in the time available. The problem becomes even more complex when NOT logic [15] is incorporated in phase fault trees. Approximations are used to obtain mission failure probabilities. An alternative to fault trees can be used, known as Binary Decision Diagrams (BDDs), that express the failure logic function in a disjoint form. This property provides an efficient quantification process that can give exact values. Due to this advantage, BDDs are identified as suitable risk assessment tool for the real-time analysis of the system failure probability that is required in the decision making process of autonomous vehicles.

4. Phased Mission Methodology for Autonomous Vehicles

This section contains a description of a simple UAV mission, the overview of the phased mission methodology and its application to the example.

4.1. System and mission description

Consider a simple UAV mission. The mission objective is to travel from airport A, perform a reconnaissance task while cruising above certain location and successfully return to airport A. A very simple view of the mission consists of three mission phases: take-off, cruise and landing. The task to be fulfilled by the UAV is considered as a part of cruise phase, since the surveillance is performed while cruising. All three phases must be successfully completed in order for the mission to be a success, i.e. the UAV performs a successful flight and completes the reconnaissance task. Each phase may only begin after the successful completion of previous phases.

Simplified phase fault trees to illustrate the concepts are shown in Figure 3. These formulate the failure logic F_i for each phase.

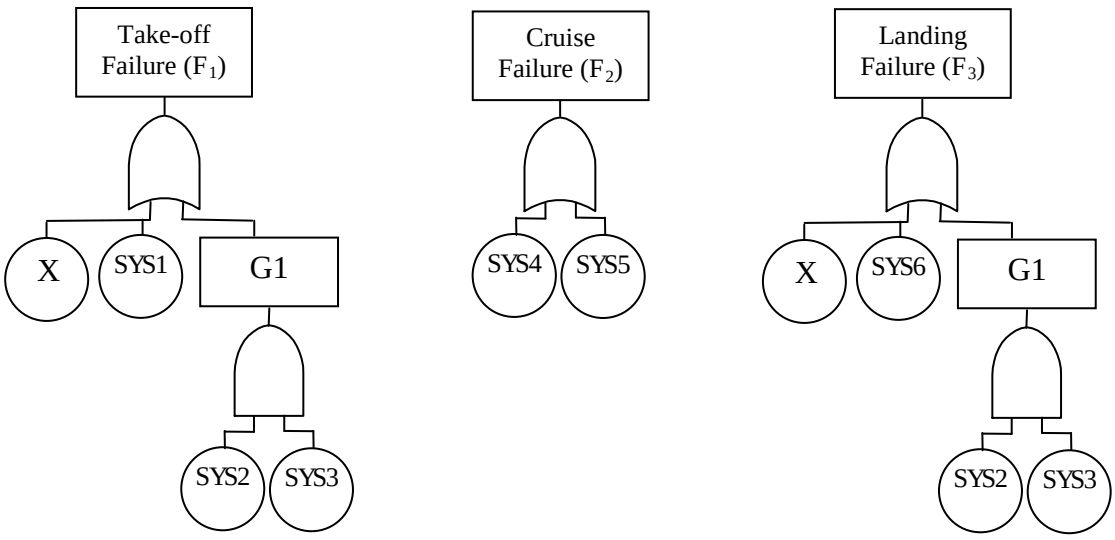

Figure 3 - Fault trees of F_1, F_2 and F_3 for the example system

The failure of take-off phase is defined as the failure of subsystem SYS1, or the failure of the two redundant subsystems SYS2 and SYS3, or the occurrence of external factor X. This factor X can be poor weather conditions, for example, strong rain or blizzard, that would prevent the take-off of the aircraft. The cruise phase fails if one of the two subsystems SYS4 or SYS5 fail, where SYS4 is needed to retain the capability to fly and SYS5 handles the surveillance capability. The landing fault tree is very similar to the take-off fault tree, where the occurrence of external factor X, the failure of subsystem SYS6 or the failure of the two redundant subsystems SYS2 and SYS3 can prevent the landing. Clearly these fault trees are very much smaller than for a real problem but are sufficient to demonstrate the concepts.

While operating the autonomous vehicle, it is important that the UAV makes rational decisions to preserve the safety of the flight and achieve the mission success in the event of component or sub-system failures or the occurrence of external events. For example, if the surveillance equipment fails, it fails the cruise phase in the current mission configuration. In the alternative configuration the UAV could be required to perform a different objective or follow a different route. The following section gives an overview of the phased mission methodology for autonomous systems and its application to the UAV example.

4.2. Mission planning methodology and its implementation

The reliability-based method and its application to the mission planning strategy is presented step-by-step:

- Convert phase fault trees, each of them representing failure logic in the phase, to BDDs
- Calculate initial phase and mission failure probabilities
- Once the new information (failures, event occurrence) is available, update phase and mission failure probabilities

- If the failure probability exceeds the defined limit, reconfiguration of the mission is considered

Each step of the methodology is explained in more detail, in the context of its application for the simple UAV example in the following sections.

4.2.1 Fault tree to BDD conversion

Once the mission is defined, the methodology starts by converting phase fault trees to BDDs. The well-known **ite** technique [4] can be used for this purpose. When large systems are considered, the conversion process can be time consuming, therefore, in order to minimise the time taken for the analysis this part of the methodology can be done off-line. In this case, the most compact BDD representation can be achieved beforehand, applying different component ordering schemes that can affect the size of the resulting BDD. Building BDDs before the start of the mission fulfils the requirements of the efficiency in the speed of analysis.

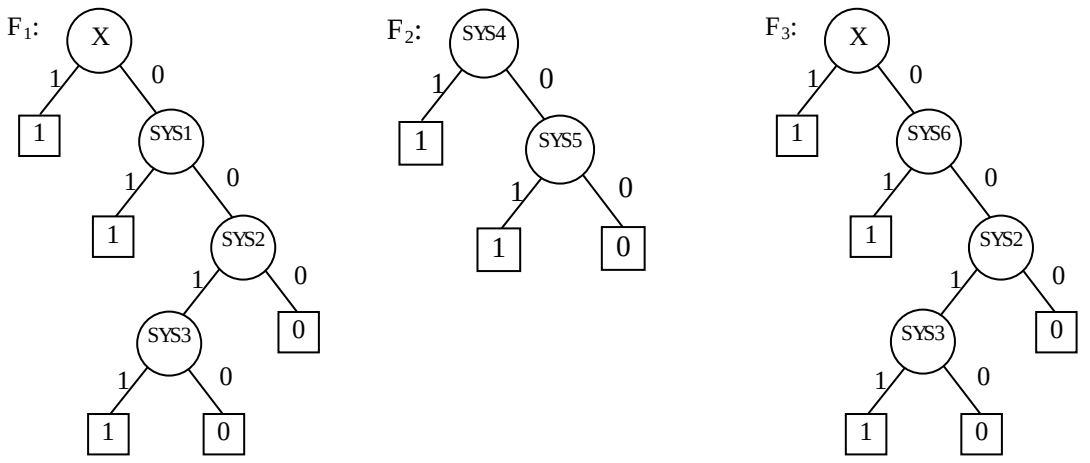

Figure 4 – BDDs of F_1, F_2 and F_3 for the example system

For example, the three orderings have been assigned for the phase fault trees, shown in Figure 3, i.e. for the take-off phase the variable ordering X < SYS1 < SYS2 < SYS3 is assigned, for the cruise phase – SYS4 < SYS5 and for the landing phase – X < SYS6 < SYS2 < SYS3. The BDDs obtained are shown in Figure 4.

These BDDs can now be used in the quantification process.

4.2.2. Calculation of initial phase and mission failure probabilities

The quantification process is based on the method presented in [11] where a detailed explanation of the method is given. The BDDs representing the failure in phase i given successful completion of the preceding phases 1,…,i-1, Ph_i, are used in this process.

First of all, the start and end times of each phase are required, the start time of phase i is denoted by t_{i-1} and the end time of phase i is denoted by t_i. The start time of phase i is equal to the end time of phase i-1. For the simplicity, assume that $t_i = i$. For example, phase I starts (as well as the whole mission) at time $t_0 = 0$ and it ends at time $t_1 = 1$.

After that, it is necessary to associate the time interval for each node, which identifies the time period over which the event can occur in order for it to contribute to phase i failure represented by the BDD of F_i. The time dependent indexes are assigned to each node in the phase BDDs, as it is shown in Figure 5.

In this representation, the first index is the start time of the mission and the second index is the end time of the current phase. This applies to system variables, denoted by SYS_i. The only external factor variable has the first index representing the start time of the current phase, but not the beginning of the mission, since external factor can independently happen in any of the

phases. The BDDs in Figure 5 encode the failure logic of each phase taking into account the time intervals over which the variables can contribute to phase failure.

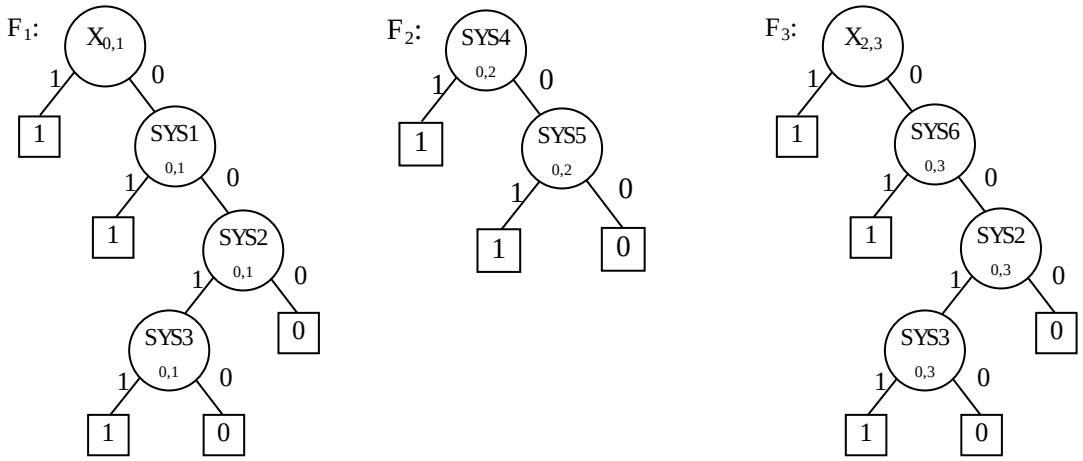

Figure 5 – BDDs of F_1, F_2 and F_3 with time dependent indexes

In the quantification process BDDs representing Ph_i are used. A BDD for Ph_i is obtained by calculating the conjunction of the BDDs representing the success in all the previous phases and the BDD of F_i for the current phase i. The BDD representing Ph_1 is equivalent to the BDD representing F_1, shown in Figure 5, since it has no preceding phases. The BDD representing Ph_2 is obtained by applying the AND operation between the two BDDs, i.e. the BDD of the success in its only one preceding phase 1 and the BDD of F_2. The first of the two BDDs is obtained from F_1 by replacing the 1 terminal nodes by the 0 terminal nodes and replacing the 0 terminal nodes by the 1 terminal nodes. Then a simple connection of the two BDDs is performed. During the AND operation between the two BDDs, the second BDD is connected on every available 1 terminal node of the first BDD. No global variable ordering is required in this process. The two resulting BDDs representing the Ph_2 and Ph_3 are shown in Figure 6.

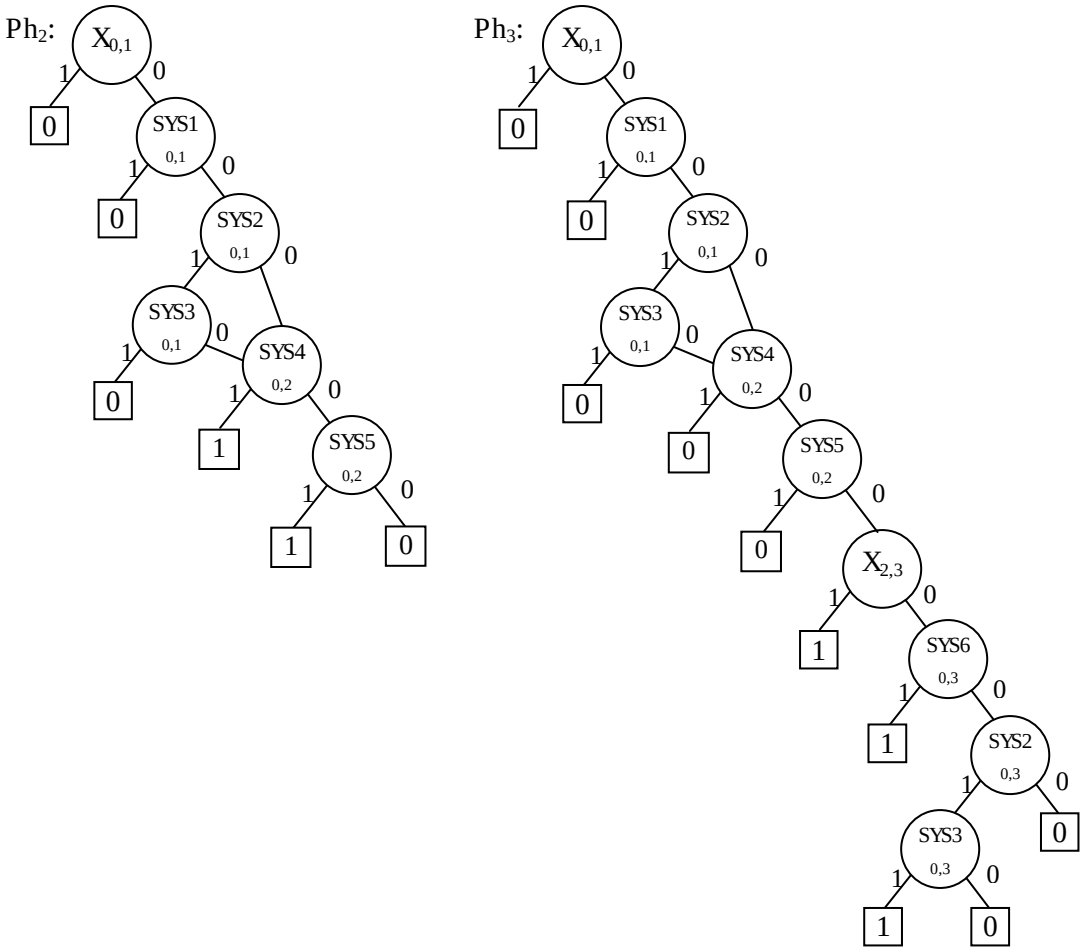

Figure 6 – BDDs of Ph_2 and Ph_3

In each of those BDDs all paths from the root vertex to terminal vertex 1 are traced in order to express the conditions of phase failure in a simplified form used in the quantification of phase and mission failure probability. Paths that cause failure of a phase are listed below.

Phase I:

1. $X_{0,1}$
2. $\overline{X_{0,1}} SYS1_{0,1}$ (7)
3. $\overline{X_{0,1}}\ \overline{SYS1_{0,1}} SYS2_{0,1} SYS3_{0,1}$

Phase II:

1. $\overline{X_{0,1}}\ \overline{SYS1_{0,1}}SYS2_{0,1}\overline{SYS3_{0,1}}SYS4_{0,2}$
2. $\overline{X_{0,1}}\ \overline{SYS1_{0,1}}SYS2_{0,1}\overline{SYS3_{0,1}}\ \overline{SYS4_{0,2}}SYS5_{0,2}$
3. $\overline{X_{0,1}}\ \overline{SYS1_{0,1}}\ \overline{SYS2_{0,1}}SYS4_{0,2}$
4. $\overline{X_{0,1}}\ \overline{SYS1_{0,1}}\ \overline{SYS2_{0,1}}\ \overline{SYS4_{0,2}}SYS5_{0,2}$

(8)

Phase III:

1. $\overline{X_{0,1}}\ \overline{SYS1_{0,1}}\ SYS2_{0,1}\overline{SYS3_{0,1}}\ \overline{SYS4_{0,2}}\ \overline{SYS5_{0,2}}X_{2,3}$
2. $\overline{X_{0,1}}\ \overline{SYS1_{0,1}}SYS2_{0,1}\overline{SYS3_{0,1}}\ \overline{SYS4_{0,2}}\ \overline{SYS5_{0,2}}\ \overline{X_{2,3}}SYS6_{0,3}$
3. $\overline{X_{0,1}}\ \overline{SYS1_{0,1}}SYS2_{0,1}\overline{SYS3_{0,1}}\ \overline{SYS4_{0,2}}\ \overline{SYS5_{0,2}}\ \overline{X_{2,3}}\ \overline{SYS6_{0,3}}SYS2_{0,3}\,SYS3_{0,3} \rightarrow$
   $\overline{X_{0,1}}\ \overline{SYS1_{0,1}}SYS2_{0,1}\,SYS3_{1,3}\overline{SYS4_{0,2}}\ \overline{SYS5_{0,2}}\ \overline{X_{2,3}}\ \overline{SYS6_{0,3}}$
4. $\overline{X_{0,1}}\ \overline{SYS1_{0,1}}\ \overline{SYS2_{0,1}}\ \overline{SYS4_{0,2}}\ \overline{SYS5_{0,2}}X_{2,3}$
5. $\overline{X_{0,1}}\ \overline{SYS1_{0,1}}\ \overline{SYS2_{0,1}}\ \overline{SYS4_{0,2}}\ \overline{SYS5_{0,2}}\ \overline{X_{2,3}}SYS6_{0,3}$
6. $\overline{X_{0,1}}\ \overline{SYS1_{0,1}}\ \overline{SYS2_{0,1}}\ \overline{SYS4_{0,2}}\ \overline{SYS5_{0,2}}\ \overline{X_{2,3}}\ \overline{SYS6_{0,3}}SYS2_{0,3}\,SYS3_{0,3} \rightarrow$
   $\overline{X_{0,1}}\ \overline{SYS1_{0,1}}SYS2_{1,3}\overline{SYS4_{0,2}}\ \overline{SYS5_{0,2}}\ \overline{X_{2,3}}\ \overline{SYS6_{0,3}}SYS3_{0,3}$

(9)

Paths 3 and 6 of phase III are simplified, removing the overlapping intervals of repeated occurrences of variables. The general rules for the simplification of paths are shown in [11] and are also described below. They simplify the time intervals associated with variables that occur more than once along the path to be quantified. This is where the dependencies between phase variables are addressed.

If system component k is considered, then:

$$x_k(t_{i1}, t_{j1}).x_k(t_{i2}, t_{j2}) = x_k(\max(t_{i1}, t_{i2}), \min(t_{j1}, t_{j2})) \quad (10)$$

where $x_k(t_i, t_j)$ is the binary indicator variable and $t_i < t_j$,

$$x_k(t_i, t_j) = \begin{cases} 1, \text{if component k fails from time } t_i \text{ to time } t_j, \\ 0, \text{otherwise.} \end{cases} \quad (11)$$

In this notation the working state of component $\overline{x_k(t_0, t_j)}$ is equivalent to $x_k(t_j, \infty)$, i.e. the fact that component k has not failed from the beginning of the mission until time t_j is equivalent to the fact that component k fails some time after t_j. In the example, equation (10) is applied resulting in the simplification $\overline{SYS3_{0,1}}SYS3_{0,3} = SYS3_{1,3}$ in expression (9).

If external factor X is considered, no simplification rules are applied, as shown in expression (9).

The phase failure probability Q_i is obtained by summing the probabilities of each path, for example, for phase I, Q_1 is calculated as shown below:

$$Q_1 = q_{X_{0,1}} + \left(1 - q_{X_{0,1}}\right)\int_{t_0}^{t_1} f_1(t)dt + \left(1 - q_{X_{0,1}}\right)\int_{t_1}^{\infty} f_1(t)dt \int_{t_0}^{t_1} f_2(t)dt \int_{t_0}^{t_1} f_3(t)dt \quad (12)$$

where $q_{X_{0,1}}$ is the failure probability of external factor X in the time period from t_0 to t_1, and f_k is the failure probability density function for sub-system SYSk.

Finally, by summing the phase failure probabilities the overall mission failure probability is obtained, i.e.

$$Q_{MISS} = Q_1 + Q_2 + Q_3. \quad (13)$$

This concludes the initial failure probability calculation.

Updating failure probabilities and reconfiguration

Once the mission is in progress, the failure probability is updated taking into account new information about the changes in the system or in the environment. The BDDs representing Ph_i are updated taking into account the current states of components or subsystems. The updated mission failure probability is then checked against the acceptable limit.

For example, after the successful completion of the take-off phase, the failure probabilities for the future phases are updated. Using equation (5) the updated probabilities are:

$$Q_{2|\bar{1}} = \frac{Q_2}{1-Q_1} \text{ and } Q_{3|\bar{1}} = \frac{Q_3}{1-Q_1} \tag{14}$$

$$Q_{MISS|\bar{1}} = Q_{2|\bar{1}} + Q_{3|\bar{1}}. \tag{15}$$

When failures occur they can affect the future phases. For example, the failure of subsystem SYS5, that ensures the surveillance capability results in the cruise phase failure. If this failure occurs in the take-off phase this phase will be completed successfully, but on entering the cruise phase, the phase failure will occur, as well as the overall mission failure. However, if this prognosis is available for the decision making process, an alternative mission objective could be achieved, avoiding to use the failed subsystem, even if this is, for this simplified mission an abandonment and return to base. As another example, if, while the UAV is in cruise phase, poor weather conditions arise, i.e. component X occurs, and the landing phase failure probability increases so much that the UAV is unable to land safely, then it either changes the course and lands in a different airport or if possible cruises while the weather conditions improve.

For simple systems like that phase fault trees are small and their analysis can be easily performed. However, it can be time consuming to convert large fault trees to BDDs,

especially if the ordering scheme is unsuitable. Also, when the number of phases in the mission increases, the number of paths for the quantification process also increases and the analysis can take too long, especially in real-time applications. A way to improve the efficiency is to reduce the size of the problem without loosing the required complexity. Therefore, phase fault trees can be reduced prior to the BDD conversion process. The modularisation method applied to phase fault trees is presented in the following section.

5. Phase fault tree modularisation

This modularisation method simplifies original fault trees, reducing the representation of the failure logic and identifying independent subtrees, whose solution is equivalent to the original fault tree. These independent modules can be analysed separately and then the results combined to give the overall result. The modularisation algorithm is presented in [9]. In the phased mission analysis this method is applied to phase fault trees, as shown below. After the modularisation is applied, the subtrees are converted to BDDs, which are analysed separately and then the results are combined to obtain the failure probability of the phase. In the phased mission analysis the subtrees, that appear in more than one phase, are converted to BDDs only once and then the expression is reused as many times as required during the quantification process. In phased mission it is common to have sub-systems whose failure affect the failure of each phase. For example, for the UAV mission an obvious example of such a system is the fuel system, whose failure would stop engines working in any phase. If a part of fault tree corresponding to a sub-system failure can be analysed independently, its size will have a smaller impact on the overall size of phase BDD, i.e. a smaller number of paths need to be visited, than during the analysis of BDDs without modules.

A module of a fault tree is a subtree that is completely independent from the rest of the tree. It contains no basic events that appear elsewhere in the fault tree. This definition from [9] is

extended for the phased mission analysis. In this case such modules are independent from the rest of the fault tree of the current phase and the other phases. As such, the module in a fault tree can be extracted only after all phase fault trees are taken into account.

The modules can be identified using the linear-time algorithm of Dutuit and Rauzy which traverses the fault trees twice. Following this algorithm gates G1 and Phase II are identified as modules. The other two gates cannot be modules since they contain repeated gates and events. The modularised phase fault trees are shown in Figure 7.

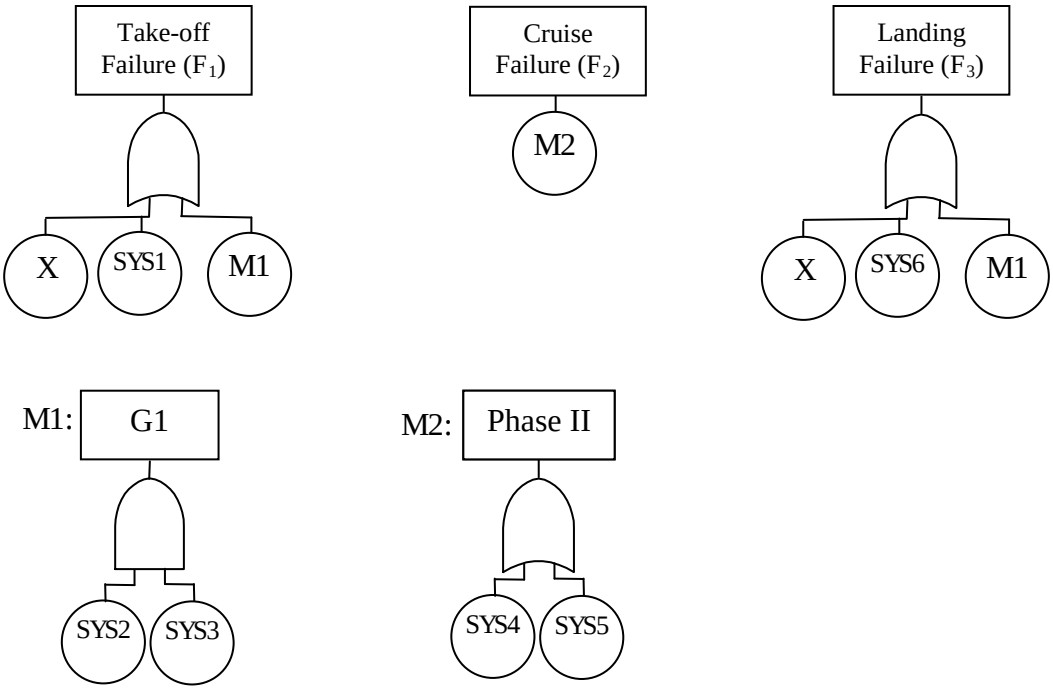

Figure 7 – Modularised phase fault trees for F_1, F_2 and F_3 and modules M1 and M2

Now the modularised fault trees are converted to BDDs, as shown in Figure 8. After that the BDDs for Ph_1, Ph_2 and Ph_3 are created in the way, described in section 4.2.2. Those BDDs are shown in Figure 9.

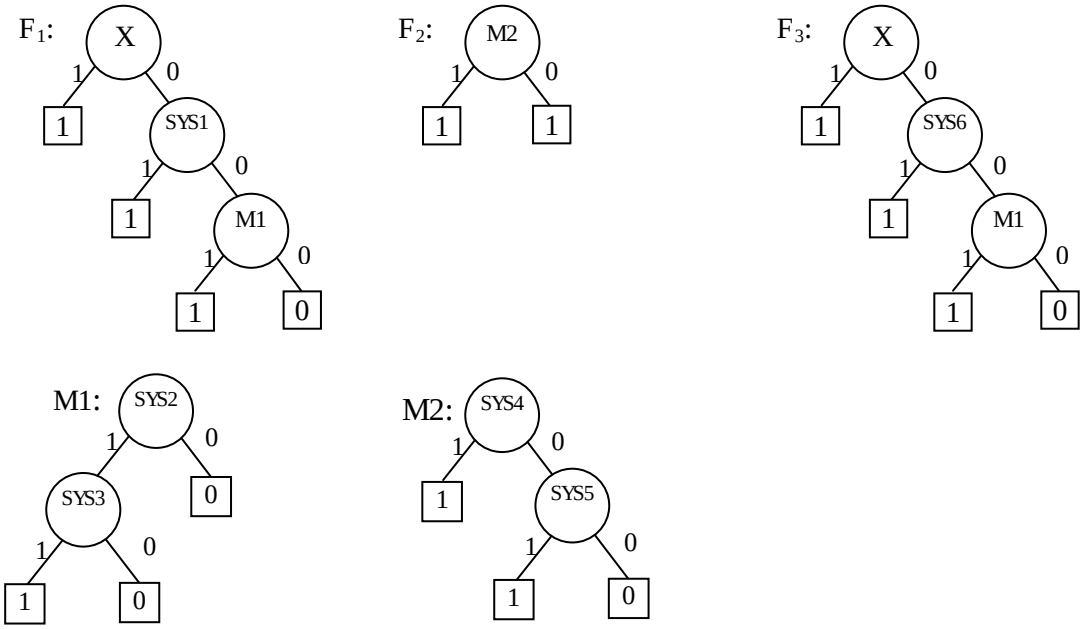

Figure 8 – BDDs for F_1, F_2 and F_3 and modules M1 and M2

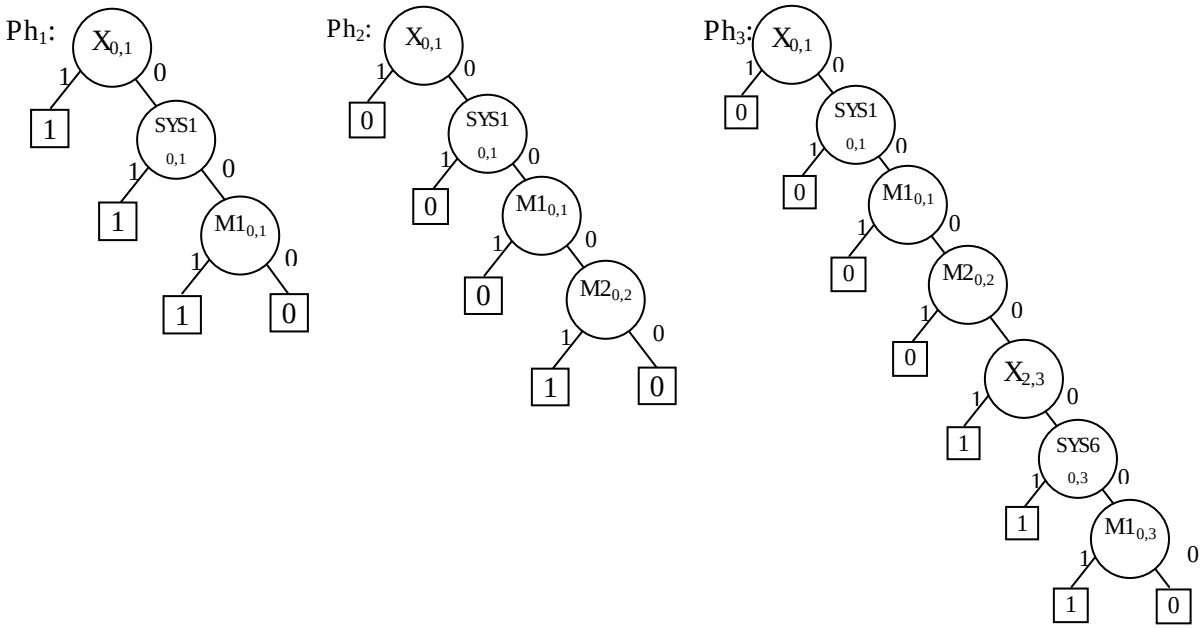

Figure 9 – BDDs of Ph_1, Ph_2 and Ph_3 and modules M1 and M2

The quantitative analysis is performed on the set of BDDs in Figure 9 to obtain the failure probability for each phase. The number of paths for the quantitative analysis is reduced using

the modularised phase fault trees instead of the original phase fault trees, as it is shown below.

A better efficiency can be achieved when the probability of a module failure is calculated only once when module appears more than once in the phase fault trees. Using single modular event in the BDD to present the failure logic of the subsystem reduces the number of paths in the quantification process. Also, the same BDD is used to calculate module failure probability in different time intervals. For example, if $q_{M1_{t_0,t_1}}$ and $q_{M1_{t_0,t_2}}$ are required, the same BDD of module M1 in Figure 8 is traversed twice for the evaluation. Calculating $q_{M1_{t_0,t_1}}$ for nodes in the BDD the time of failure is taken from time t_0 to time t_1, and calculating $q_{M1_{t_0,t_2}}$ the time of failure is taken from time t_0 to time t_2. Then those probabilities can be used to obtain the module failure probabilities in different interval, i.e.

$$q_{M_{t_i,t_j}} = q_{M_{0,t_j}} - q_{M_{0,t_i}} . \tag{16}$$

This increases the efficiency of the analysis, since the structure of the module does not need to be visited again.

For example, for phase III there are only 3 paths using the modularised fault trees in comparison to 6 paths using the original fault trees.

Phase III:

1. $\overline{X_{0,1}}\ \overline{SYS1_{0,1}}\ \overline{M1_{0,1}}\ \overline{M2_{0,2}}\ X_{2,3}$

2. $\overline{X_{0,1}}\ \overline{SYS1_{0,1}}\ \overline{M1_{0,1}}\ \overline{M2_{0,2}}\ \overline{X_{2,3}}\ SYS6_{0,3}$

3. $\overline{X_{0,1}}\ \overline{SYS1_{0,1}}\ \overline{M1_{0,1}}\ \overline{M2_{0,2}}\ \overline{X_{2,3}}\ \overline{SYS6_{0,3}}\ M1_{0,3} \rightarrow$

$\overline{X_{0,1}}\ \overline{SYS1_{0,1}}\ M1_{1,3}\ \overline{M2_{0,2}}\ \overline{X_{2,3}}\ \overline{SYS6_{0,3}}$. (17)

In this case, $q_{M1_{0,1}}$ can be calculated traversing the BDD of module M1 and then reusing it as many times as needed. Also, the same BDD can be used to calculate the probability of module failure at any other time during the mission. As shown in equation (16):

$$q_{M1_{1,3}} = q_{M1_{0,3}} - q_{M1_{0,1}}. \tag{18}$$

Therefore, the phase III failure probability is calculated as:

$$\begin{aligned} Q_1 = {} & p_{X_{0,1}}\, p_{SYS1_{0,1}}\, p_{M1_{0,1}}\, p_{M2_{0,2}}\, (q_{X_{2,3}} + p_{X_{2,3}}\, q_{SYS6_{0,3}}) \\ & + p_{X_{0,1}}\, p_{SYS1_{0,1}}\, q_{M1_{1,3}}\, p_{M2_{0,2}}\, p_{X_{2,3}}\, p_{SYS6_{0,3}}. \end{aligned} \tag{19}$$

Finally, using equation (4) the overall mission failure probability is obtained.

The efficiency of the modularisation technique in the phased mission analysis was tested using an example UAV mission that consists of 13 phases and the complexity of phase fault trees is given in Table 1.

Phase	Number of gates	Number of basic events	Number of modules
I	1	3	0
II	19	20	6
III	20	19	6
IV	18	20	6
V	18	18	6
VI	20	21	6
VII	19	19	6
VIII	18	17	6
IX	19	20	6
X	19	19	6
XI	18	21	6
XII	18	17	6
XIII	1	2	0

Table 1 – UAV mission for the efficiency analysis

Phase fault trees are medium sized, number of gates and number of events given in columns 2 and 3 respectively. Column 4 shows number of modules identified in each phase. During the

test number of nodes in the phase BDDs, number of paths to quantify and time taken to perform the analysis were recorded and given in Table 2, column 1, 2 and 3 respectively.

	Number of nodes	Number of paths	Time of analysis
Without modularisation	5359	15212924	12.36
With modularisation	2807	116452	0.03
Decrease (%)	47.62	99.24	99.76

Table 2 – Test results using the non-modularised and modularised phase fault trees

The efficiency of the analysis improved a lot after modules were identified. The number of paths was reduced dramatically, since some complex parts of a BDD were replaced by a single node representing a modular event. Due to this, the speed of the analysis has increased a lot.

The efficiency of the modularisation technique in the phased mission modelling depends on how common modules are in phase fault trees. It is likely that a system undergoing a phased mission will have some subsystems whose failure contributes to each phase failure in the same way, as it was discussed in the example above. Therefore, the identification of independent modules of those contributions and their implementation in the method proposed, increases the speed of the analysis. This improvement is especially important if the methodology is required to support the decision making process in real time.

6. Conclusions

A reliability-based methodology employing BDDs has been developed for the phased mission analysis. It forms the base of a prognostics tool used in a mission planning strategy of autonomous vehicles. The methodology takes into account changing environmental

conditions and failures that occur on the system. The updated phase and mission failure likelihood can support real-time decision making process. The speed of the analysis is improved by employing the modularisation technique, where smaller BDDs are obtained and can be analysed quickly.

The phased mission methodology can be applied as a prognostics tool in the mission planning of autonomous vehicles. This contribution is particularly important when the mission failure probability is revaluated throughout the mission according to fault diagnostics information and can be used as a part of the decision making strategy. The methodology has been successfully demonstrated on the ASTRAEA project.

7. Acknowledgements

The authors would like to acknowledge the financial support for this work provided by DTI as part of ASTRAEA programme.

8. References

[1] Esary JD, Ziehms H. Reliability analysis of phased missions. In: Barlow RE, Fussell JB, Singpurwalla ND, editors. Reliability and Fault Tree Analysis: Theoretical and Applied Aspects of System Reliability and Safety Assessment. Philadelphia, PA: Society for Industrial and Applied Mathematics; 1975:213-36.

[2] Kohda T, Wada M, Inoue K. A simple method for phased mission analysis. Reliability Engineering and System Safety. 1994(45):299-309.

[3] Ma Y, Trivedi KS. An algorithm for reliability analysis of phased mission systems. Reliability Engineering and System Safety. 1999(66):157-70.

[4] Rauzy A. New algorithms for fault tree analysis. Reliability Engineering and System Safety. 1993(40):203-11.

[5] Sinnamon RM, Andrews JD. Improved accuracy in quantitative fault tree analysis. Quality and Reliability Engineering International. 1997(13):285-92.

[6] Sinnamon RM, Andrews JD. Improved efficiency in qualitative fault tree analysis. Quality and Reliability Engineering International. 1997(13):293-98.

[7] Zang X, Sun H, Trivedi KS. A BDD-based algorithm for reliability analysis of phased-mission systems. IEEE Transactions on Reliability. 1999(48):50-60.

[8] LaBand R, Andrews JD. Phased mission modelling using fault tree analysis. Proceedings of the IMechE, Part E. Journal of Process Mechanical Engineering. 2004(218):83-91.

[9] Dutuit Y, Rauzy A. A linear-time algorithm to find modules of fault trees. IEEE Transactions on Reliability. 1996(45):422-5.

[10] Reay KA, Andrews JD. A fault tree analysis strategy using binary decision diagrams. Reliability Engineering and System Safety. 2002(78):45-56.

[11] Prescott DR, Remenyte-Prescott R, Reed S, Andrews JD, Downes CG. A reliability analysis method using BDDs in phased mission planning. Proceedings of the IMechE, Part O: Journal of Risk and Reliability. 2009(223):133-43.

[12] Andrews JD. A ternary decision diagram method to calculate the component contribution to the failure of systems undergoing phased missions. Proceedings of the IMechE, Part O: Journal of Risk and Reliability. 2008(222):173-87.

[13] Prescott DR, Remenyte-Prescott R, Andrews JD. A systems reliability approach to decision making in autonomous multi-platform missions operating a phased mission. In: Proceedings of the Annual Reliability and Maintainability Symposium, Las Vegas, USA, January 2008:8-14.

[14] Vesely WE. A time dependent methodology for fault tree evaluation. Nuclear Design and Engineering. 1970(13):337-60.

[15] Andrews JD. The use of not logic in fault tree analysis. Quality and Reliability Engineering International. 2001(17):143-50.