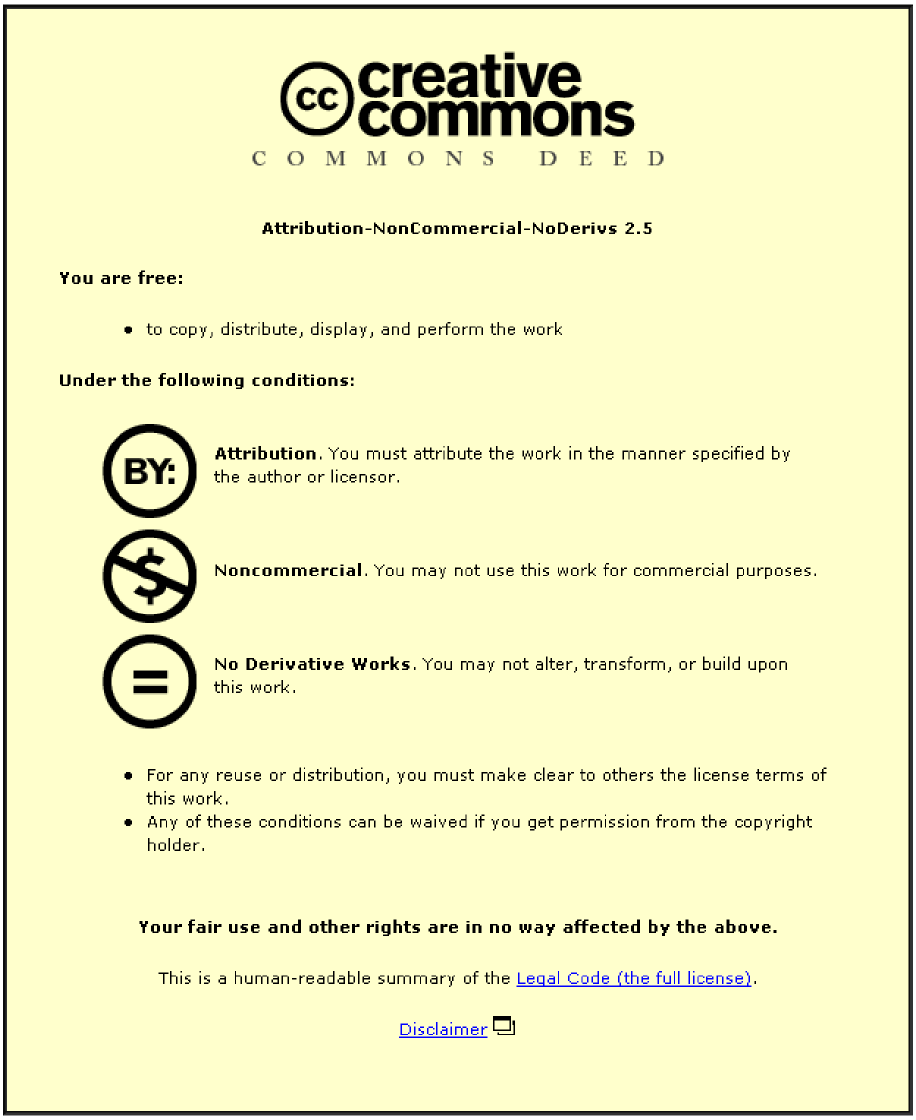

A reliability analysis method using binary decision diagrams in phased mission planning

D R Prescott[1]*, R Remenyte-Prescott[1], S Reed[1], J D Andrews[1], and C G Downes[2]
[1]Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, UK
[2]BAE Systems, Preston, UK

The manuscript was received on 24 June 2008 and was accepted after revision for publication on 3 December 2008.

DOI: 10.1243/1748006XJRR202

Abstract: The use of autonomous systems is becoming increasingly common in many fields. A significant example of this is the ambition to deploy unmanned aerial vehicles (UAVs) for both civil and military applications. In order for autonomous systems such as these to operate effectively, they must be capable of making decisions regarding the appropriate future course of their mission responding to changes in circumstance in as short a time as possible. The systems will typically perform phased missions and, owing to the uncertain nature of the environments in which the systems operate, the mission objectives may be subject to change at short notice. The ability to evaluate the different possible mission configurations is crucial in making the right decision about the mission tasks that should be performed in order to give the highest possible probability of mission success.

Because binary decision diagrams (BDDs) may be quickly and accurately quantified to give measures of the system reliability it is anticipated that they are the most appropriate analysis tools to form the basis of a reliability-based prognostics methodology. The current paper presents a new BDD-based approach for phased mission analysis, which seeks to take advantage of the proven fast analysis characteristics of the BDD and enhance it in ways that are suited to the demands of a decision-making capability for autonomous systems. The BDD approach presented allows BDDs representing the failure causes in the different phases of a mission to be constructed quickly by treating component failures in different phases of the mission as separate variables. This allows flexibility when building mission phase failure BDDs because a global variable ordering scheme is not required. An alternative representation of component states in time intervals allows the dependencies to be efficiently dealt with during the quantification process. Nodes in the BDD can represent components with any number of failure modes or factors external to the system that could affect its behaviour, such as the weather. Path simplification rules and quantification rules are developed that allow the calculation of phase failure probabilities for this new BDD approach. The proposed method provides a phased mission analysis technique that allows the rapid construction of reliability models for phased missions and, with the use of BDDs, rapid quantification.

Keywords: phased mission analysis, autonomous systems, decision-making strategy, binary decision diagrams

1 INTRODUCTION

A phased mission is one in which a system is required to perform a number of different tasks or functions in sequence. The periods in which each of these successive tasks or functions takes place are known as phases. Each phase must be completed successfully in order that the mission can be considered a success. Many systems operate such phased missions, with aircraft being a prime example. When performing an unreliability assessment for phased mission systems, failure probabilities are calculated for each of the individual phases and these are used

Corresponding author: Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough LE11 3TU, Leicestershire, UK. email: d.r.prescott@lboro.ac.uk to obtain the mission failure probability for the entire mission.

Systems whose components can undergo repair during a phased mission are known as repairable phased mission systems, whereas systems for which repair are not permitted during the mission are known as non-repairable phased mission systems. Markov techniques can deal with the dependencies introduced when repairable phased mission systems are analysed [**1**]. Fault tree techniques are suitable when modelling non-repairable phased mission systems as the absence of repair processes facilitates the use of such techniques that require independence [**2**]. Non-repairable phased missions are considered in the present paper.

The size and complexity of many systems means that exact fault tree quantification is impossible and upper bounds are often used to approximate the failure probabilities. For this reason it is common to convert fault trees to binary decision diagrams (BDDs) before quantification takes place. The conversion process requires variables, representing fault tree basic events, to follow a specified ordering scheme and can be time-consuming for large fault trees. However, advantages are gained when converting fault trees to BDDs, because the structure of BDDs allows system failure quantification to be carried out quickly and accurately, with exact solutions being obtained without the need for approximations [**3**].

The BDD approach has been applied to phased mission analysis for non-repairable systems in reference [**4**]. A fault tree is constructed, taking into account the success of previous mission phases, which represents mission failure in each phase under consideration, as well as the overall mission failure probability. In order to do this a basic event transformation, detailed in reference [**2**], is used, which replaces all basic events in a phase fault tree by a number of basic events that represent occurrence of each of the basic events in each mission phase. This results in a considerable increase in the number of basic events to be included in the analysis. The fault tree representing mission failure in the phase under consideration is then converted to a BDD, which is quantified to give an exact mission phase failure probability. In order to construct this BDD a global variable ordering scheme must be specified that encapsulates all variables in the previous successful phases as well as the phase currently considered for mission failure. This variable ordering scheme can have a significant effect on the size of the resultant BDD and, as a consequence, will affect the time taken to perform the quantification process. In reference [**5**] a similar global variable ordering scheme is required, which then allows quantification of the overall mission failure probability, with a phase algebra being used to deal with dependencies across phases. Thus, in methods such as those in references [**4**] and [**5**], if an unsuitable ordering scheme is chosen, a lengthy process of BDD construction can occur prior to quantification.

In the phased mission analysis method presented in reference [**6**] a notation is introduced for components that explicitly expresses the time periods over which components work or fail. This is used to obtain the mission failure conditions and calculate the mission failure probability. However, the method presented requires that minimal cut sets and minimal path sets be obtained, using fault tree analysis (FTA), in advance, for each phase. This requires considerable, perhaps impractical, computational effort.

This paper takes advantage of the rapid quantification offered by BDDs and incorporates the basic event time period notation used in reference [**6**] in order to provide the opportunity quickly to construct BDDs for the analysis of mission phases for a system. This is implemented in order to facilitate the use of the decision making strategy presented in reference [**7**]. A brief overview of BDDs and phased mission analysis is followed by the BDD methodology proposed to meet the challenge of providing a real-time prognostics capability within a decision-making strategy. It is also described how events with multiple failure modes and external factors are incorporated in the methodology.

2 BACKGROUND

2.1 Binary decision diagrams

BDDs provide an alternative approach to fault trees to represent the failure logic of a system. BDDs can be used for the accurate quantification [**8**, **9**] of fault trees, because exact solutions can be calculated without the requirement to evaluate minimal cut sets as an intermediate step. This method improves the accuracy and efficiency of conventional approaches [**10**] and is proving to be of considerable use in system reliability analysis.

2.1.1 Binary decision diagram definition

A BDD is a directed acyclic graph, where all paths through the BDD start at the root vertex and terminate in one of two states: a 1-state (system failure) or a 0-state (system success). The BDD is composed of terminal and non-terminal vertices, which are connected by branches. Terminal vertices correspond to the final state of the system and non-terminal vertices correspond to the basic events of the fault tree. By connection, all left branches leaving a vertex are the 1-branches (component fails), all right branches

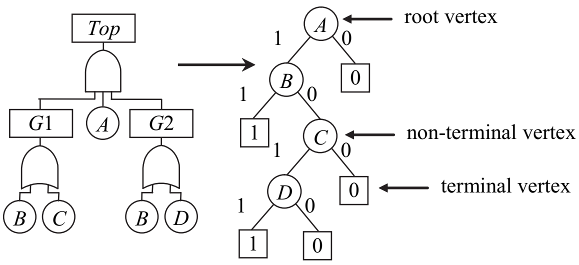

Fig. 1 Example fault tree converted to BDD

are the 0-branches (component functions). The construction of the BDD requires the basic events to be ordered. In Fig. 1 an example fault tree is converted to a BDD with the ordering $A < B < C < D$. The ordering scheme employed can have a considerable effect on the number of nodes in the BDD, particularly for large fault trees. For this reason, it is important to select an ordering scheme that will minimize the size of the BDD and hence allow fast quantification.

Each path that terminates in a 1-state gives a cut set, i.e. a combination of component failure conditions where the existence of all of them will result in system failure, when only failed states of components are considered. For example, following the first path in Fig. 1 gives a cut set {A, B}.

The BDD encodes the logic function of the system failure in its disjoint form, therefore, the probability of occurrence of the top event, Q_{SYS}, can be expressed as the sum of the probabilities of the disjoint paths through the BDD. Because paths through the BDD are mutually exclusive, the probability of failure for the system in Fig. 1, Q_{SYS}, is expressed as

$$Q_{\mathrm{SYS}} = q_{\mathrm{A}}q_{\mathrm{B}} + q_{\mathrm{A}}(1 - q_{\mathrm{B}})q_{\mathrm{C}}q_{\mathrm{D}} \tag{1}$$

where q_i represents the probability of failure of component i.

2.1.2 *Binary decision diagram construction*

A commonly used method of constructing BDDs was developed by Rauzy [**3**]. This approach applies an if-then-else (ite) technique to each of the gates in the fault tree. If $f(x)$ is the Boolean function for the top event then the given ite structure ite (x, f_1, f_2) means that if variable x occurs (fails) then consider f_1, else consider f_2, where f_1 and f_2 are Boolean functions, known as the residues of f, with $x = 1$ and $x = 0$ respectively. Therefore, in the BDD structure f_1 lies below the 1-branch of the node encoding x and f_2 lies below the 0-branch.

First of all, a variable ordering for basic events needs to be established. Then the conversion of every gate to the BDD is performed according to the following rules. For gates whose inputs have already been defined as an ite structure the rule of the conversion process is applied, i.e. if $J = \mathrm{ite}(x, f_1, f_0)$ and $H = \mathrm{ite}(y, g_1, g_0)$ represent two inputs to a gate of logic type $\oplus$, then

$$J \oplus H = \begin{cases} \mathrm{ite}(x, f_1 \oplus H, f_0 \oplus H) & \text{if } x < y \text{ in the ordering,} \\ \mathrm{ite}(x, f_1 \oplus g_1, f_0 \oplus g_0) & \text{if } x = y \text{ in the ordering.} \end{cases} \tag{2}$$

The resulting BDD shown in Fig. 1 is an ordered BDD, where traversing the BDD along any path from the root vertex will encounter the nodes in the order specified. Using this approach the variable ordering is retained throughout the BDD because every step of the connection is performed according to the ordering of the elements. Also, the method automatically uses subnode sharing, storing each ite structure in the memory only once and reusing calculated ite structures further in the process.

2.2 Phased mission analysis

A phased mission defined in this paper has the following characteristics:

(a) a mission contains a number of consecutive and sequential phases;
(b) a specified task has to be accomplished in each phase and therefore there are different failure criteria in each phase;
(c) for a mission to be successful all phases must be completed successfully;
(d) the duration of each phase is known;
(e) all components are working before the start of the mission;
(f) the mission is non-repairable and component failures remain once they have occurred.

The phased mission is represented by a number of fault trees, each of them expressing the conditions leading to the failure of a particular phase. A method of calculating the mission failure probability is detailed in reference [**4**]. The method works by calculating the probability of failure, Q_i, in each of the mission phases, i, and then adding these to give the total mission failure probability, Q_{MISS}.

For any phase the method combines the causes of success of previous phases with the causes of failure for the phase being considered. The general fault tree for this is shown in Fig. 2. As can be seen from the diagram, in order for the mission to fail in phase i, the failure conditions must not have been met in any of the previous $i - 1$ phases and then the failure conditions for phase i must be met. Let F_j represent the logical expression for the failure conditions

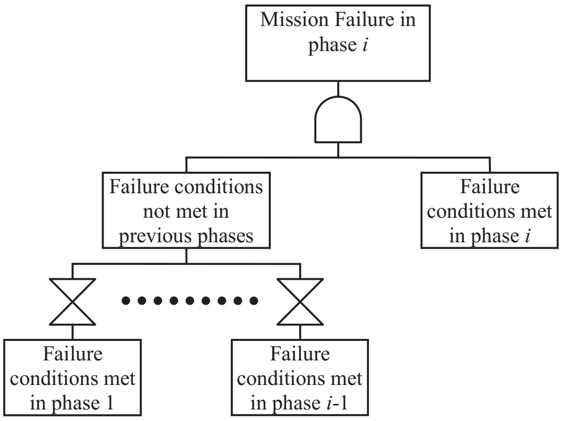
Mission Failure in phase i
Failure conditions not met in previous phases
Failure conditions met in phase i
Failure conditions met in phase 1
Failure conditions met in phase i-1

Fig. 2 Fault tree for mission failure in phase i

being met in phase j and Ph_j represent the logical expression for mission failure in phase j. Then

$$\begin{aligned} Ph_1 &= F_1 \\ Ph_2 &= \overline{F_1} \cdot F_2 \\ &\vdots \\ Ph_j &= \overline{F_1} \cdot \overline{F_2} \cdot \overline{F_3} \cdot \ldots \cdot \overline{F_{j-1}} \cdot F_j \end{aligned} \tag{3}$$

These logical expressions are represented by fault trees such as that in Fig. 2. FTA can be used to quantify the probability of mission failure during mission phase i

$$Q_i = P(Ph_i) \tag{4}$$

The total mission failure probability is given by adding these mission phase failure probabilities

$$Q_{\text{MISS}} = \sum_{i=1}^{n} Q_i \tag{5}$$

When efficiency and accuracy of the analysis is important, the fault trees for mission failure in phase i, representing Ph_i, are converted to BDDs to obtain the probability of mission failure accurately. This means that the logical expressions for mission failure in phase i presented in equation (3) are effectively converted to a BDD format, which allows fast and efficient quantification of the mission phase failure probabilities.

3 AUTONOMOUS SYSTEM MISSION PLANNING METHODOLOGY

3.1 Motivation

The motivation behind this research is to develop a strategy for the reliability analysis of phased missions which could be used as part of a decision-making capability for autonomous systems. In reference [**7**] a decision-making strategy for autonomous systems is presented, for which a phased mission analysis capability is required. Because results from this phased mission analysis capability are required in order to make decisions as to how the phased mission being performed should progress, it is imperative that reliable results can be obtained in as short a time as possible.

For example, consider an unmanned aerial vehicles (UAV) that is required to perform whichever of a number of possible missions is most likely to be completed successfully. Consider also that it must autonomously make a decision as to which mission to perform in a short period of time, as the window of opportunity for performing each of the missions is small. A technique is required that allows the UAV quickly to quantify the probability of success for each of the possible missions in order to decide which mission to perform. The chosen mission will be the one with the highest probability of success.

The methodology presented here is aimed at performing a phased mission analysis in order to quantify the probability, Q_i, of system failure in each of the i mission phases of a particular mission being performed and also the probability of failure over the course of the entire mission, Q_{MISS}.

3.2 Overview

BDDs offer the potential to move towards the real-time quantitative phased mission analysis demanded by a prognostics capability in a decision-making strategy. They allow fast quantification of the mission phase failure probabilities, Q_i, which can then be used to calculate the total mission failure probability, Q_{MISS}. However, considerable time can be spent converting fault trees to BDDs, time which would severely impact on the ability to offer real-time analysis of phased mission systems. For this reason, the phased mission modelling methodology presented is focussed on reducing this construction time and having phased mission BDDs available as early as possible in order that analysis can begin soon after a mission configuration becomes known. Essentially, the idea is that certain parts of the methodology are carried out offline, before a mission configuration is known, and other parts online, once the mission configuration is known. The goal is to minimize the online processes that must take place before quantification can begin and therefore move towards a real time analysis that can be used in a decision-making process. The idea is that in starting the quantification sooner it may be apparent at an earlier time whether or not the failure probability of a certain mission configuration is acceptable.

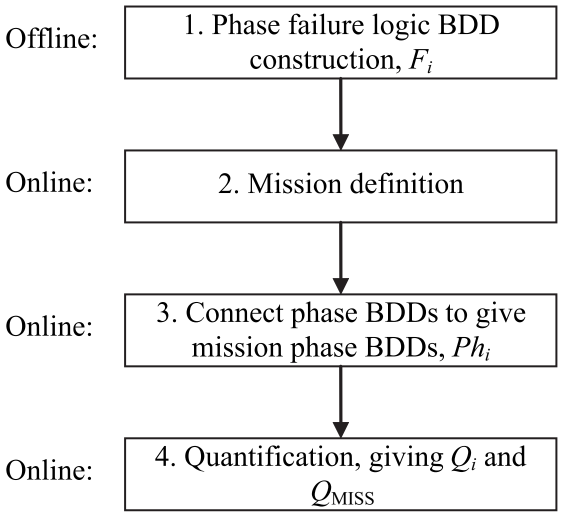
Offline:
1. Phase failure logic BDD construction, F_i
Online:
2. Mission definition
Online:
3. Connect phase BDDs to give mission phase BDDs, Ph_i
Online:
4. Quantification, giving Q_i and Q_{MISS}

Fig. 3 Stages of the methodology

Figure 3 gives a representation of the steps involved in the phased mission analysis. These are briefly described below.

1. This step can be carried out offline, before a mission configuration is known. For any system to which the decision-making capability will be applied, the failure of all possible tasks (mission phases) that can possibly be performed must be represented using fault trees. These are then converted to the BDDs that will be used to represent the failure conditions being met in each of the mission phases, F_i. The methodology allows these BDDs to be constructed independently and thus time can be taken in choosing a suitable variable ordering scheme for each BDD that enables its size to be minimized. These BDDs will then effectively be stored in a library ready for later use. Since this step is carried out offline, as much time as is available can be taken to perform this step.
2. This is the first of the online steps, and will contribute towards the time taken to perform an analysis after a phased mission becomes known. The mission is defined in terms of a mission profile. This specifies the order of the tasks that the system is to perform and the time to be taken doing each of them. This information is applied to the appropriate BDDs from the library, resulting in BDDs representing F_i, which are thus ready to be used to construct the BDDs for mission failure in the various mission phases, Ph_i. The time taken to perform this step will be minimal since it is a very simple process, as will be described later.
3. This step is crucial in being able to begin quantitative analysis as quickly as possible. Rather than having to combine the BDDs representing F_i and follow a global variable ordering scheme for all phase variables, where dependencies between variables must be taken into account, in constructing BDDs representing Ph_i, there is a simple connection process that will take little time to perform. It requires no further variable ordering and allows rapid connection of the F_i BDDs, each of which follows its own variable ordering scheme that was assigned during step 1 in order to attempt to minimize its own size.
4. Quantification of the Ph_i BDDs can begin. The calculated failure probabilities can be monitored during the quantification process for acceptability.

In the following sections the BDD based phased mission methodology is described in more detail. It is also described how a number of categories of variable can be dealt with using the methodology.

3.3 Notation

In the proposed methodology there are three categories of variable that will be considered. Two of these categories are components of the system and the final one represents factors external to the system that influence its operation during the mission. Descriptions of each of these follow, along with the notation that will be used for each.

3.3.1 Single failure mode variables

These variables represent components of the system that can exist in only two states, either working or failed. Each component has an indicator variable, $x_k(t_i, t_j)$, defined as follows

$$x_k(t_i, t_j) = \begin{cases} 1, & \text{if component } k \text{ fails} \\ & \text{from time } t_i \text{ to time } t_j \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

where $k = 1,\ldots,n_S$ and n_s is the number of single failure mode components in the system.

When considering the success state of single failure mode variables this equivalence holds

$$\overline{x_k}(t_0, t_i) = x_k(t_i, \infty) \tag{7}$$

This means that when component k does not fail from time t_0 until time t_i, it can fail after time $t_{i,}$, where t_0 is the start time of the mission.

3.3.2 Multiple failure mode variables

These variables represent components of the system that can exist in a working state or one of a number of known failed states. For example, consider a valve that can fail in an open or a closed position. Each of these components has an indicator variable defined as

$$x_k^l(t_i, t_j) = \begin{cases} 1, \text{ if component } k \text{ fails} \\ \quad \text{from time } t_i \text{ to time } t_j \text{ in mode } l, \\ 0, \text{ otherwise.} \end{cases} \tag{8}$$

where $k = 1, \ldots, n_{\mathrm{M}}$, n_{M} being the number of multiple failure mode components in the system, $l = 1, \ldots, m$, with m being the number of failure modes of component k.

When considering multiple failure mode variables in their success state, a similar equivalence to the one for single failure mode variables holds

$$\overline{x_k^l}(t_0, t_i) = x_k^l(t_i, \infty) \tag{9}$$

This means that when component k does not fail from time t_0 until time t_i in its failure mode l, it can fail in that failure mode after time t_i.

3.3.3 External factor variables

These variables represent factors that would appear in a system's phase fault tree but would not be a part of the system. Examples of such factors are an electrical storm or rain that could affect the performance of an aircraft. The indicator variable for these factors is defined as

$$x_k^e(t_i, t_j) = \begin{cases} 1, \text{ if external factor } k \text{ appears} \\ \quad \text{from time } t_i \text{ to time } t_j, \\ 0, \text{ otherwise.} \end{cases} \tag{10}$$

where $k = 1, \ldots, n_{\mathrm{E}}$ and n_{E} is the number of external factors.

3.4 Methodology steps

The methodology involves a number of steps, given that fault trees are known for all of the possible phases that can be performed by the system. These fault trees will represent certain tasks or functions that the system can perform. A number of these are configured in sequence in order to fulfil the requirements of a mission. Note that, as the methodology is described here, the mission configuration for the system is initially unknown, i.e. the ordering and length of phases is not yet determined. The steps in the methodology are described in the following subsections. Each step will be applied to an example system, which can perform three possible tasks, represented by the fault trees shown in Fig. 4.

A number of basic events appear in the fault trees for the example system: basic events A, B, C, and D represent system components and basic event X is an external factor that influences the behaviour of the system. All system components, except A, are single failure mode components, and component A can fail in two modes, i.e. A^1 and A^2.

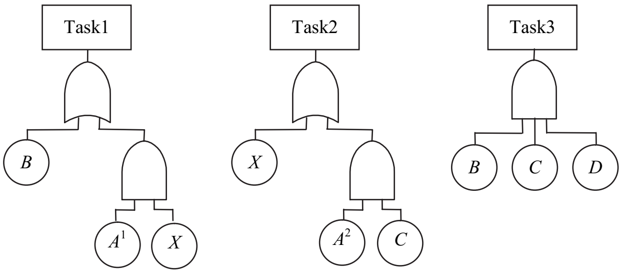

Fig. 4 A library of fault trees for an example system

3.4.1 Convert system phase fault trees to BDDs

In order to be ready to evaluate the probability of mission failure in as short a time as possible when the mission configuration is decided the fault trees for the potential mission phases, i.e. the tasks that can be performed by the system, are converted to BDDs using the techniques outlined earlier. This means that the time taken to construct the BDDs does not impinge on the time available to quantify them once the mission configuration is decided. Each BDD is converted using its own variable ordering scheme, which is chosen in order to minimize, as much as possible, the size of the BDD. The variables of the BDDs will each fall into one of the three categories outlined above (single or multiple mode failure or external factor).

Performing this step on the example system requires that a variable ordering be assigned to each fault tree before the fault trees are converted to BDDs. Using a simple top–down left–right traversal of basic events in the fault trees for tasks 1, 2, and 3 sets the variable ordering schemes, i.e. for Task1 $-B < A^1 < X$, for Task2 $-X < A^2 < C$ and for Task3 $-B < C < D$. The BDDs obtained are shown in Fig. 5. This library of BDDs is now ready to be used in the phased mission analysis as soon as the mission configuration is known.

3.4.2 Mission definition and variable time association

When the mission is defined for the system the tasks or functions that must be performed, and the sequence in which they should occur, are known. The time at which each task or function will start and end is also known and this means that the phases of the mission have now been determined. Given this information it is now necessary to assign the time interval over which each of the variables contributes to phase failure for each of the phase failure logic

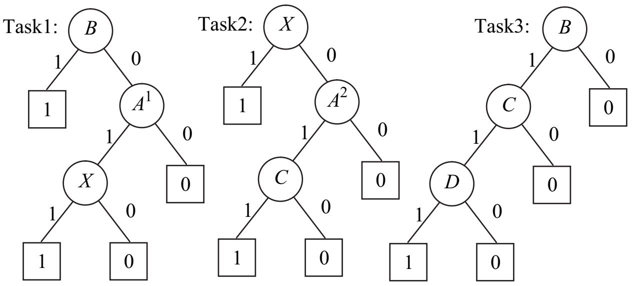

Fig. 5 The library of BDDs representing example system tasks

BDDs representing F_i. Thus, each indicator variable, as defined in equations (6), (8), and (10) will now have an associated t_i and t_j. These will be determined as follows for the three variable categories.

1. For the single failure mode and multiple failure mode variables: t_i is given by the start time for the mission and t_j by the end time for the phase with which the variable is associated. This is because the failure of components represented by these variables can contribute to the failure of the phase at any time during the period from the start of the mission to the end of that phase.
2. For the external factor variables: t_i and t_j are given by the start and end times respectively of the phase with which the variable is associated. This is attributable to the fact that whether or not the external factor occurred before the phase is not important to the failure of this phase. All that matters is whether or not the external factor occurs in that phase. For example, if one considers rain affecting a system in a certain phase, it will not matter if the rain occurred at an earlier time, only if it occurs in the phase under consideration. If this assumption is not true, and, for example, rain in a phase could affect the performance of the system in a later phase, then the external factor must be treated as a single failure mode variable. In this case time t_i and t_j are given by the start and end times respectively of the interval when the external factor affects the system performance.

Consider the case where the example system will perform a mission consisting of three phases, where phase I is represented by Task1, phase II by Task3, and phase III by Task2. Given this mission configuration, BDDs representing F_1, F_2, and F_3 are selected from the BDD library to be used in the subsequent phased mission analysis. Start and end times are assigned to each phase as shown in Table 1. Thus the time intervals over which each of the variables will contribute to phase failure is known and thus each node in the BDDs selected for the mission phases is assigned two time indices as shown in Fig. 6. The BDDs in Fig. 6 now encode the failure logic of each phase, taking into account the time intervals over which the variables can contribute to phase failure. That is, not only for the current phase but also for preceding phases, if appropriate. For example, for the single failure mode variables in the phase 2 BDD (F_2) the state of the components in phases 1 and 2 is taken into account. The dependencies between related variables in different phases are taken account of during quantification.

Table 1 Phase start and end times

Phase	Start time	End time
I	$t_0=0$	$t_1=1$
II	$t_1=1$	$t_2=2$
III	$t_2=2$	$t_3=3$

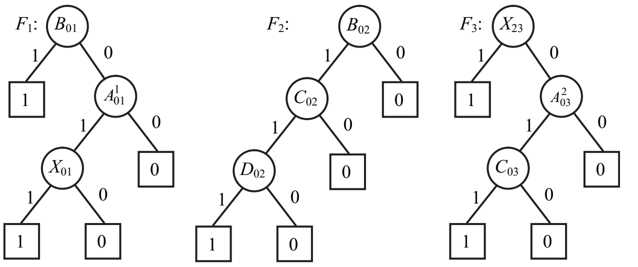

Fig. 6 Phase BDDs representing failure conditions being met in phases 1, 2, and 3, i.e. F_1, F_2, and F_3

3.4.3 *Connection of phase failure logic BDDs*

This step of the methodology involves building the logical expressions for mission failure in phase i, Ph_i, represented in equation (3) by using the appropriate BDDs for the failure conditions being met in phase i, F_i. When using BDDs to represent F_i and times are associated with the variables as described above the process of constructing BDDs, representing F_i is relatively simple. In order to consider the success of the mission in a certain phase the 0 and 1 terminal nodes of the BDD are swapped. This gives a dual BDD representing success in a phase. The AND connection of two BDDs, performed when building the Ph_i BDDs, is done by connecting all terminal 1 nodes of one BDD to the root node of the BDD to be connected to that BDD. Although different BDDs might contain identical variables that would normally be required to adhere to a specific ordering scheme covering both BDDs, this is not the case with this method. Instead, upon connection the variables

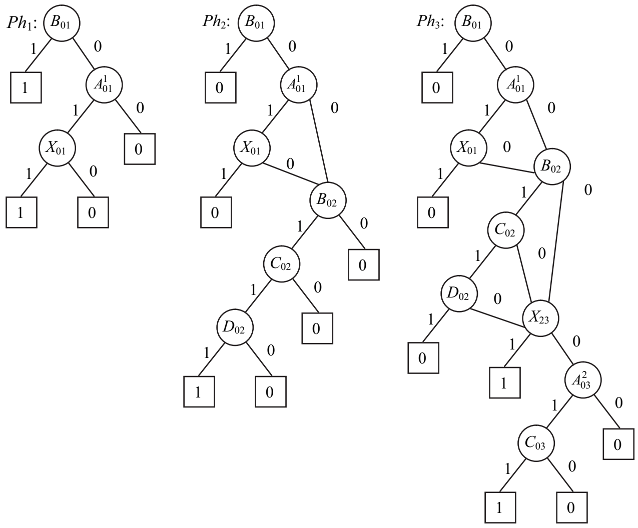

Fig. 7 BDDs representing Ph_1, Ph_2, and Ph_3

are treated as independent, with the times associated with the variables being used to take into account dependencies between them during quantification. This vastly reduces time taken to construct the mission phase failure BDDs representing Ph_i, allowing fast, efficient connection of the phase BDDs representing F_i.

For the example mission the BDDs representing Ph_i, obtained after using the rules above to connect the BDDs representing F_i as required to represent previous phase success and current phase failure, are shown in Fig. 7. These BDDs are now ready to be quantified to obtain the mission phase failure probabilities.

3.4.4 *Phase failure quantification*

In this step of the methodology the Ph_i BDDs constructed by connecting the phase failure logic BDDs, F_i, are quantified to give the mission phase failure probabilities, as in equation (4). The quantification process involves tracing through the BDD from the root node along all possible paths to terminal 1 nodes. Traversing the 1 branch from a node corresponds to component failure or occurrence and traversing along the 0 branch corresponds to component success or non-occurrence. When considering the success state related to a variable equations (7) and (9) are used. Since all paths are disjoint, the path probabilities are added to give the total mission phase failure probability. A general path in the BDD to be quantified will contain more than one instance of a variable owing to the fact that a global variable ordering scheme for the BDDs was not required. Each instance represents the same component failure in different time intervals. Taking account of these time intervals allows dependencies between related variables in different phases to be considered. In order to allow quantification to take place a process of simplifying the path failure logic must take place as the path is traced through the BDD. This process works by performing simple calculations with the times associated with each variable encountered on the path, applying simplification rules described below. In this way, upon reaching a terminal 1 node, every variable encountered along the specific path leading to that node will have associated with it a start and an end time, dependencies between variables will have been removed and quantification may take place. The time intervals within which the variables occur (given by the associated start and end times) govern the quantification process that occurs for each variable on the path as the terminal 1 node is reached. The process for the three different types of variable is given below. Once all of the probabilities have been calculated for the individual variables along the path to be quantified they are multiplied together to give the path probability.

1. *Single failure mode variables*

Simplification rules applied to single failure mode variables are defined as follows

$$x_k(t_i, t_j) = 0, \text{ if } t_i > t_j \tag{11}$$

$$\begin{aligned} x_k(t_{i1}, t_{j1}).x_k(t_{i2}, t_{j2}) = {} & x_k(\max(t_{i1}, t_{i2}), \\ & \min(t_{j1}, t_{j2})) \end{aligned} \tag{12}$$

Applying these rules to a path leads to a variable $x_k(t_i,\ t_j)$ which may or may not be equal to 0. If not, then a probability for this variable is calculated as follows

$$P\left[x_k\left(t_i, t_j\right) = 1\right] = \int_{t_i}^{t_j} f_k(t)\mathrm{d}t \tag{13}$$

where $f_k(t)$ is the failure probability density function for component k.

This probability can then be used in the quantification of a path containing component k.

2. *Multiple failure mode variables*

Two cases are presented for multiple failure mode variables, the first being when only one failure mode appears on a path and the second when more than one failure mode appears on a path.

If there is only one failure mode (out of all the possible failure modes) considered on a path equivalent rules to equations (11) and (12) are used

$$x^l_k(t_i, t_j) = 0, \text{ if } t_i > t_j \tag{14}$$

$$x_k^l(t_{i1}, t_{j1}).x_k^l(t_{i2}, t_{j2}) = x_k^l(\max(t_{i1}, t_{i2}), \min(t_{j1}, t_{j2})) \tag{15}$$

If two different failure modes, l_1 and l_2, are considered on a path $l_1 \neq l_2$, equation (12) is expressed as follows

$$x_k^{l1}(t_{i1}, t_{j1}).x_k^{l2}(t_{i2}, t_{j2}) = 0 \tag{16}$$

This is attributed to the fact that a component cannot fail in more than one failure mode. Equation (16) is applied if $t_{j1} \neq \infty$ and $t_{j2} \neq \infty$.

If $t_{j1} \neq \infty$ and $t_{j2} \neq \infty$, $x_k^{l1}(t_{i1}, t_{j1})$ is expressed as $\overline{x_k^{l1}}(t_0, t_{i1})$, using equation (9). Then equation (16) is expressed as given below

$$\overline{x_k^{l1}}(t_0, t_{i1}).x_k^{l2}(t_{i2}, t_{j2}) = x_k^{l2}(t_{i2}, t_{j2}) \tag{17}$$

This rule is explained by the fact that if a component has failed in a particular failure mode l_2 then it cannot have failed in any other failure mode.

Applying these rules to a path containing components with multiple failure modes yields a variable $x_k^l(t_i, t_j)$which may or may not be equal to 0. If not, then a probability for this variable is calculated in two cases.

(a) *Case 1*: Only one of the possible failure modes for a variable appears in the simplified path logic.

In this case the failure probability for the variable is calculated in the same way that it was for the single failure mode variable, i.e.

$$P\left[x_k^l\left(t_i, t_j\right) = 1\right] = \int_{t_i}^{t_j} f_k^l(t)dt \tag{18}$$

where $f_k{}^l$ is the failure probability density function for component k in failure mode l.

(b) *Case 2*: More than one of the possible failure modes for a variable appears in the path logic.

It only occurs in situations when working states of multiple failure mode variables are considered, since the combinations of more than one failed states of the same variable have been removed using equation (16). When this occurs the failure probability for the variable is calculated as follows

$$\begin{aligned} & P\left[x_k^{l1}(t_{i1}, \infty) \cdot \ldots \cdot x_k^{lm}(t_{im}, \infty) = 1\right] \\ & = P\left[x_k^{l1}(t_0, t_{i1}) + \ldots + x_k^{lm}(t_0, t_{im}) = 0\right] \\ & = 1 - P\left[x_k^{l1}(t_0, t_{i1}) + \ldots + x_k^{lm}(t_0, t_{im}) = 1\right] \end{aligned} \tag{19}$$

Since $x_k^{l1}(t_0, t_{i1}) \ldots x_k^{lm}(t_0, t_{im})$ are mutually exclusive, i.e. component k cannot fail in more than one failure mode at the same time, this expression is equivalent to

$$\begin{aligned} & 1 - \left\{P\left[x_k^{l1}(t_0, t_{i1}) = 1\right] + \ldots + P\left[x_k^{lm}(t_0, t_{im}) = 1\right]\right\} \\ & = 1 - \left(\int_{t_0}^{t_{i1}} f_k^{l1} \mathrm{d}t + \ldots + \int_{t_0}^{t_{im}} f_k^{lm} \mathrm{d}t\right) \end{aligned} \tag{20}$$

3. *External factors*

For variables that represent external factors no rules for the simplification of paths need to be considered, because external factors occur independently in each phase. Quantification takes account of the mission phases under consideration

$$P\left[x_k^e\left(t_i, t_j\right) = 1\right] = q_k^e\left(t_i, t_j\right) \tag{21}$$

$$P\left[\overline{x}_k^e\left(t_i, t_j\right) = 1\right] = 1 - q_k^e\left(t_i, t_j\right) \tag{22}$$

The BDDs representing Ph_i for the example system shown in Fig. 7 are traversed from the top node to each terminal 1 vertex and the paths are identified. For phase I there are two paths.

1. B_{01}
2. $B_{1\infty}A_{01}^1X_{01}$

No simplification rules need be applied to these phase 1 paths, because no components are represented by more than one variable in each path. In addition to this, for the variable representing the component with multiple failure modes, A_{01}^1, this also appears in isolation in the path.

For simplicity use $q_{k_{ij}^l}$, which describes the failure probability of component k failing in mode l in time interval (t_i, t_j). For example, $q_{A_{01}^1}$ is the probability of component A failing in its first failure mode in time interval (t_0, t_1). Using this terminology and the quantification rules above the phase I failure probability, Q_1, is calculated as

$$Q_1 = q_{B_{01}} + (1 - q_{B_{01}})q_{A_{01}^1}q_{X_{01}}^e$$

For phase II there also are two paths, shown below on the left. Here the simplification rule (12) is applied to a single failure mode variable $B_{1\infty}B_{02} = B_{12}$ leading to the two simplified expressions of the path logic shown on the right.

1. $B_{1\infty}A_{01}^1\overline{X_{01}}B_{02}C_{02}D_{02}$ 1. $B_{12}A_{01}^1\overline{X_{01}}C_{02}D_{02}$

2. $B_{1\infty}A_{1\infty}^1B_{02}C_{02}D_{02}$ 2. $B_{12}A_{1\infty}^1C_{02}D_{02}$

The phase II failure probability Q_2 is calculated as

$$\begin{aligned} Q_2 = \; & q_{B_{12}}q_{A_{01}^1}(1 - q_{X_{01}}^e)q_{C_{02}}q_{D_{02}} \\ & + q_{B_{12}}(1 - q_{A_{01}^1})q_{C_{02}}q_{D_{02}} \end{aligned}$$

Considering phase III there are 12 paths in the BDD for Ph_3, shown on the left following.

1.$B_{1\infty}A^1_{01}\overline{X_{01}}B_{02}C_{02}D_{2\infty}X_{23}$	1. $B_{12}A^1_{01}\overline{X_{01}}C_{02}D_{2\infty}X_{23}$
2. $B_{1\infty}A^1_{01}\overline{X_{01}}B_{02}C_{02}D_{2\infty}\overline{X_{23}}A^2_{03}C_{03}$	2. –
3. $B_{1\infty}A^1_{01}\overline{X_{01}}B_{02}C_{2\infty}X_{23}$	3. $B_{12}A^1_{01}\overline{X_{01}}C_{2\infty}X_{23}$
4. $B_{1\infty}A^1_{01}\overline{X_{01}}B_{02}C_{2\infty}\overline{X_{23}}A^2_{03}C_{03}$	4. –
5. $B_{1\infty}A^1_{01}\overline{X_{01}}B_{2\infty}X_{23}$	5. $B_{2\infty}A^1_{01}\overline{X_{01}}X_{23}$
6. $B_{1\infty}A^1_{01}\overline{X_{01}}B_{2\infty}\overline{X_{23}}A^2_{03}C_{03}$	6. –
7. $B_{1\infty}A^1_{1\infty}B_{02}C_{02}D_{2\infty}X_{23}$	7. $B_{12}A^1_{1\infty}C_{02}D_{2\infty}X_{23}$
8. $B_{1\infty}A^1_{1\infty}B_{02}C_{02}D_{2\infty}\overline{X_{23}}A^2_{03}C_{03}$	8. $B_{12}A^2_{03}C_{02}D_{2\infty}\overline{X_{23}}$
9. $B_{1\infty}A^1_{1\infty}B_{02}C_{2\infty}X_{23}$	9. $B_{12}A^1_{1\infty}C_{2\infty}X_{23}$
10. $B_{1\infty}A^1_{1\infty}B_{02}C_{2\infty}\overline{X_{23}}A^2_{03}C_{03}$	10. $B_{12}A^2_{03}C_{23}\overline{X_{23}}$
11. $B_{1\infty}A^1_{1\infty}B_{2\infty}X_{23}$	11. $B_{2\infty}A^1_{1\infty}X_{23}$
12. $B_{1\infty}A^1_{1\infty}B_{2\infty}\overline{X_{23}}A^2_{03}C_{03}$	12. $B_{2\infty}A^2_{03}\overline{X_{23}}C_{03}$

For these paths a number of simplification rules is applied. For single failure mode variables equation (12) is used to simplify $B_{1\infty}B_{02}=B_{12}$, $B_{1\infty}B_{2\infty}=B_{2\infty}$, $C_{02}C_{03}=C_{02}$, and $C_{2\infty}C_{03}=C_{23}$. For multiple failure mode variables equations (15), (16), and (17) are applied to give $A^1_{01}A^2_{03} = 0$ and $A^1_{1\infty}A^2_{03} = \overline{A^1_{01}}A^2_{03} = A^2_{03}$. Therefore, paths 2, 4, and 6 are seen to be zero and the others now have simplified path logic as shown on the right.

The phase III failure probability, Q_3, can now be calculated in the same way that Q_1 and Q_2 were calculated, quantifying the probability of each path by multiplying probabilities of each component and then summing the individual path probabilities. The overall mission failure probability can then be calculated using equation (5), i.e.

$$Q_{\mathrm{MISS}} = Q_1 + Q_2 + Q_3.$$

4 CONCLUSIONS

This paper presents a novel methodology for modelling non-repairable phased missions using BDDs. The methodology is particularly suitable for forming the basis of a prognostics capability in a decision-making strategy for autonomous systems. While the method in its current state is not capable of providing real-time analysis, the concepts used will assist in moving towards this goal. Implementing parts of the method offline and parts online allows quantitative analysis to be started as soon as possible after a mission configuration becomes known. The quantification process involves path simplification and quantification rules that will make up the greater part of the time spent in the online implementation of the methodology. The quantification process is therefore the greatest contributor to this online implementation with the time taken increasing with the complexity of the systems and number of phases in the mission. A benefit of using this technique is that there is little time taken in constructing the mission BDD model online, regardless of the complexity of the systems or number of mission phases. The methodology involves:

(a) fault tree conversion to BDDs in advance of any knowledge of mission configuration;
(b) a simple method, using component time requirements, of representing the phase of the mission in which a task is performed, which can then be used during quantification to account for phase variable dependencies;
(c) simple connection of the BDDs representing the logical expressions for failure conditions being met in phase *i* when constructing the logical expressions for mission failure in phase *i*: there is no need for a global ordering scheme to be applied;
(d) the potential to minimize the size of the mission phase failure BDDs to be quantified, taking

advantage of the fact that a global ordering scheme need not be followed and instead that phase failure logic BDDs may be individually minimized.

The methodology also includes rules that govern how components with multiple failure modes and variables that represent effects external to the system can be incorporated into the BDD analysis.

ACKNOWLEDGEMENTS

Financial support for this work has been provided by the Department of Trade and Industry (DTI), Engineering and Physical Sciences Research Council (EPSRC), and BAE SYSTEMS as part of the ASTRAEA (Autonomous Systems Technology Related Airborne Evaluation and Assessment) and NECTISE (Network Enabled Capability Through Innovative Systems Engineering) research programmes.

REFERENCES

1 **Smotherman, M.** and **Zemoudeh, K.** A non-homogeneous Markov model for phased mission reliability analysis. *IEEE Trans. Reliability*, 1989, **38**, 585–590.

2 **Esary, J. D.** and **Ziehms, H.** Reliability analysis of phased missions. *Reliability and Fault Tree Analysis*, 1975, 213–236.

3 **Rauzy, A.** New algorithms for fault tree analysis. *Reliability Engng System Safety*, 1993, **40**, 203–211.

4 **LaBand, R.** and **Andrews, J. D.** Phased mission modelling using fault tree analysis. *Proc. Instn Mech. Engrs Part E: J. Process Mechanical Engineering*, 2004, **218**, 83–91.

5 **Zang, X., Sun, H.,** and **Trivedi, K. S.** A BDD-based algorithm for reliability analysis of phased-mission systems. *IEEE Trans. Reliability*, 1999, **48**, 50–60.

6 **Kohda, T., Wada, M.,** and **Inoue, K.** A simple method for phased mission analysis. *Reliability Engng and System Safety*, 1994, **45**, 299–309.

7 **Prescott, D. R., Remenyte-Prescott, R.,** and **Andrews, J. D.** 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, 2008, pp. 8–14 (IEEE).

8 **Sinnamon, R. M.** and **Andrews, J. D.** Improved accuracy in quantitative fault tree analysis. *Quality and Reliability Engng Int.*, 1997, **13**, 285–292.

9 **Sinnamon, R. M.** and **Andrews, J. D.** Improved efficiency in qualitative fault tree analysis. *Quality and Reliability Engng Int.*, 1997, **13**, 293–298.

10 **Vesely, W. E.** A time dependent methodology for fault tree evaluation. *Nuclear Des. Engng*, 1970, **13**, 337–360.