Multiplatform phased mission reliability modelling for mission planning

D R Prescott[1]*, J D Andrews[1], and **C G Downes[2]**
[1]Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, Leicestershire, UK
[2]BAE Systems, Warton, Preston, Lancashire, UK

The manuscript was received on 26 June 2008 and was accepted after revision for publication on 17 October 2008.

DOI: 10.1243/1748006XJRR204

Abstract: Autonomous systems are being increasingly used in many areas. A significant example is unmanned aerial vehicles (UAVs), regularly being called upon to perform tasks in the military theatre. Autonomous systems can work alone or be called upon to work collaboratively towards common mission objectives. In this case it will be necessary to ensure that the decisions enable the progression of the platform objectives and also the overall mission objectives.

The motivation behind the work presented in this paper is the need to be able to predict the failure probability of missions performed by a number of autonomous systems working together. Such mission prognoses can assist the mission planning process in autonomous systems when conditions change, with reconfiguration taking place if the probability of mission failure becomes unacceptably high.

In a multiplatform phased mission a number of platforms perform their own phased mission that contributes to an overall mission objective. Presented in this paper is a methodology for calculating the phase failure probabilities of a multiplatform phased mission. These probabilities are then used to find the total mission failure probability. Prior to the mission the failure probabilities are used to decide if the original mission structure is acceptable. Once underway, failure probabilities, updated as circumstances change, are used to decide whether a mission should continue. Circumstances can change owing to failures on a platform, changing environmental conditions (weather), or the occurrence of unforeseen external events (emerging threats). This diagnostics information should be used to ensure that the updated failure probabilities calculated take into account the most up-to-date system information possible. Since the speed of decision making and the accuracy of the information used are essential, binary decision diagrams (BDDs) are utilized to form the basis of a fast, accurate quantification process.

Keywords: phased mission analysis, reliability-based prognostics, mission planning, binary decision diagrams (BDDs)

1 INTRODUCTION

Many systems perform missions consisting of several phases. Such phased missions are characterized by sequential, ordered phases, each of which must be completed in order for the mission to be successful. The requirements placed upon the system differ from phase to phase. As such, the causes of failure for each mission phase also differ. A typical example of a phased mission is an aircraft flight: taxi to runway, take-off, climb, cruise, descend, land, taxi to terminal. Platforms in the military arena also perform phased missions. The consequences of failure for these phased missions can be high and for this reason it is important to be able to analyse the reliability of these missions accurately. A number of methods

**Corresponding author: Department of Aeronautical and Automotive Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK. email: d.r.prescott@lboro.ac.uk*

are available that can be used to produce the failure probabilities of such missions. Examples are fault tree analysis [**1**], cause–consequence analysis [**2**], binary decision diagrams [**3**, **4**], Markov analysis [**5**], simulation [**6**], Petri nets [**7**], or a combination of approaches, e.g. combinatorial and Markov approaches, as in reference [**8**].

For some mission types, such as search and rescue, individual platforms are required to work collaboratively in order to achieve an overall mission objective. The move towards network-enabled capability (NEC) or network centric warfare (NCW) in military environments is also an illustration of this. Such multiplatform phased missions are characterized by the fact that individual platforms will each perform their own phased missions, within which certain tasks will contribute to the overall mission goal. In these multiplatform phased missions there is no requirement for sequential ordering of tasks, which may be carried out in parallel by different platforms, despite the sequential phases that must be performed by the single platforms. Individual platforms may not be required to be successful throughout all of the phases of their own mission in order for the mission objective to be accomplished. Successful operation up to the end of the last of their phases that contributes to the mission objective may be sufficient.

Autonomous systems that are required to operate as part of multiplatform phased missions must, by definition, make decisions about the actions that they are to perform, without human input. In systems such as these the reliability of the phased missions could form part of a decision-making strategy [**9**]. In other systems a measure of the reliability of a multiplatform phased mission could inform human decision makers as to how a mission should proceed. In either case there are two key points in a mission when a measure of the failure probability of a multiplatform phased mission system could be used. The first of these is before the mission begins. This would provide some information that could be used to assist in deciding whether or not the mission should begin in the proposed configuration. The second point is during the mission, when diagnostics information informs of a change in state of some contributor to the mission, and a new measure of the failure probability would help decide how the mission should continue.

This paper presents a novel methodology for the reliability analysis of multiplatform phased mission systems. The methodology assumes that the platforms performing the mission are non-repairable. Methods of using diagnostic data (which report the status of components, functions, or subsystems of the various platforms) in calculating the failure probability of the system while the mission is in progress are detailed, and a simple example is used to demonstrate the methodology.

2 BACKGROUND

In this section a methodology that is used for single platform phased mission analysis is outlined. This method is later extended to form part of the multi-platform phased mission methodology. Also presented is a binary decision diagram (BDD) representation that will also be used within the methodology.

2.1 Single-platform phased mission analysis

A method of calculating mission failure probabilities for a single-platform phased mission is presented in reference [**3**]. For a platform, p, the probability of mission failure, $Q_{p,i}$, in each of the mission phases, i, is added to give the total probability of failure for the platform mission, $Q_{p,\mathrm{MISS}}$. The probability of mission failure in each of the mission phases takes into account the success of previous mission phases.

The logical expressions for failure conditions being met in each of the platform phases, given by $F_{p,i}$, are combined in such a way that they give the logical expression for mission failure in phase i, $Ph_{p,i}$. This process is illustrated by the fault tree shown in Fig. 1 and takes into account the fact that in order for a platform to fail in a particular phase it must not have failed in any of the preceding phases. Thus $Ph_{p,i}$ is given by

$$Ph_{p,i} = \overline{F_{p,1}} \cdot \overline{F_{p,2}} \cdot \overline{F_{p,3}} \cdots \overline{F_{p,i-1}} \cdot F_{p,i} \tag{1}$$

The expressions given in equation (1) allow the probability of failure during mission phase i, $Q_{p,i}$, to be found. Thus the total mission failure probability is given by

$$Q_{p,\mathrm{MISS}} = \sum_{i=1}^{n} Q_{p,i} \tag{2}$$

Once the mission is in progress, updated values of the phase failure probabilities can be calculated using a method shown in reference [**10**]. Bayes' theorem is used to take into account the fact that k phases have been successfully completed and $Q_{p,j|\overline{k}}$, the failure of platform p in phase j given the successful completion of k previous phases, is given by

$$Q_{p,j|\overline{k}} = \frac{Q_{p,j}}{1 - \sum_{i=1}^{k} Q_{p,i}} \tag{3}$$

$Q_{p,\mathrm{MISS}}$ is the sum of the phase failure probabilities of the mission phases still to be completed

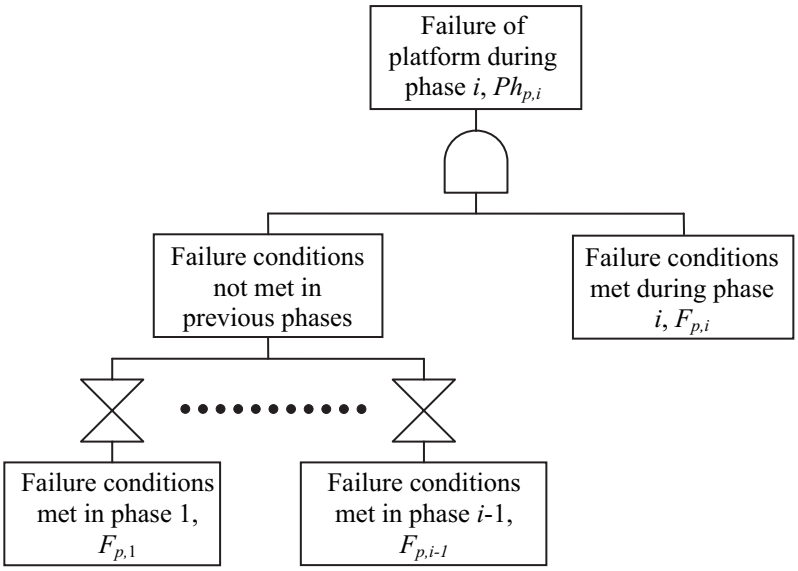


Fig. 1 Single-platform mission failure in phase i

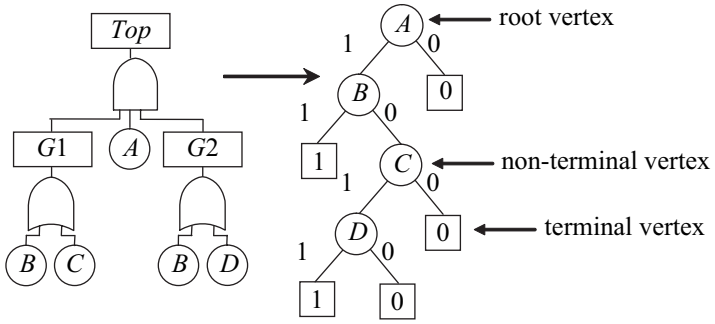


Fig. 2 A fault tree and its equivalent BDD

$$Q_{p,\mathrm{MISS}} = \sum_{j=k+1}^{n} Q_{p,j|\overline{k}} \tag{4}$$

In cases where the failure probabilities must be calculated quickly and accurately (which will be true when the probabilities will be used as part of a decision-making process), fast, accurate methods of calculation must be used.

2.2 Binary decision diagrams

A binary decision diagram (BDD) is a directed acyclic graph which can be used to encode the failure logic of a system exactly. It contains a number of paths from a root vertex to terminal 0 vertices (representing system success) and terminal 1 vertices (representing system failure). Each path is traced along 1-branches (to the left of vertices) and 0-branches (to the right), and all vertices in the BDD are ordered in such a way that the variables they represent appear in the same order no matter which path is followed. An example is shown in Fig. 2. Note that if terminal 0 and 1 vertices are swapped then the BDD representing system success is easily obtained from that representing system failure. This is of particular importance when considering the logical expressions for mission failure during a phase given by equation (1), where success of previous phases must be considered. When connecting two BDDs according to AND logic the terminal 1 vertices of one of the BDDs are replaced by the root vertex of the other BDD. When connecting using OR logic the terminal 0 vertices are replaced in the same way. In each case a global ordering scheme must be followed. Hence, any variables occurring in both BDDs must be ordered according to that ordering scheme.

A commonly used method of converting fault trees to BDDs is given in reference [**11**]. The method ensures that the specified variable ordering is followed and is also efficient in that if nodes appear in more than one place within the BDD structure they are shared. Fault tree analysis relies on the assumption that the basic events of the fault tree are independent. This assumption of independence also applies to the variables of a BDD. Hence BDDs

cannot deal with dependence between variables such as standby systems or strict failure sequences.

When a multiplatform phased mission is being conducted and decisions need to be made, whether as part of a mission performed by autonomous vehicles or controlled by human operators, the speed with which the failure probability of the mission can be provided will be of paramount importance. BDDs provide a means to obtain the failure probability for systems quickly and accurately. When modelling single-platform phased missions BDDs exactly encode the failure logic expressions for phases represented in equation (1), allowing exact quantification of the phase failure probabilities in equation (2).

A BDD approach to phased mission analysis is presented in reference [**12**] which allows the failure expressions for $Ph_{p,i}$ (in equation (2)) to be quickly constructed in BDD form by rapidly connecting the BDDs for each $F_{p,i}$. This means that the quantification process may begin soon after a mission configuration is defined. An example of the approach follows, and is illustrated in Fig. 3. It involves associating with variables in the BDDs representing the logical expressions for the phase failure conditions being met, $F_{p,i}$, the time intervals over which the variables contribute to phase failure. These time intervals are then taken account of during the BDD quantification process.

Here there are two fault trees representing the logical expressions for the failure conditions of platform 1 being met in phase 1, $F_{1,1}$, and phase 2, $F_{1,2}$, of a mission. These are converted to BDDs using an algorithm such as that suggested in reference [**11**]. Each variable of each BDD is then assigned the time interval over which that variable can cause the appropriate failure condition to be met. Thus when considering $F_{1,2}$, for example, if A and C occur from time 0 to the end of the second phase the failure conditions for phase 2 will be met. The resulting BDDs are shown in the intermediate steps in Fig. 3, after the phase failure fault trees and before the BDD representing mission failure in phase 2, $Ph_{1,2}$. These BDDs now represent the failure of the platform to complete its first and second phases. However, when obtaining a dual BDD that represents successful completion of a particular phase, the terminal 1s and 0s are simply swapped, since 1 represents mission failure and 0 represents mission success. Therefore the BDD representing mission failure in phase 2, $Ph_{1,2}$, can be constructed by swapping terminal 0 and 1 vertices of the $F_{1,1}$ BDD (thus representing successful operation through phase 1) and forming the AND combination with the $F_{1,2}$ BDD (by connecting terminal 1 nodes of NOT $F_{1,1}$ to the root node of $F_{1,2}$). Note that since the variables of $F_{1,1}$ and $F_{1,2}$ have different time intervals associated with them they are treated as independent when BDD connection occurs (hence no ordering scheme must be obeyed for variable A). The quantification algebra deals with such dependencies between variables when paths are traced.

2.3 BDD quantification

Quantification of the BDDs representing mission failure in the different phases takes place by tracing along all paths of the BDD that lead to terminal 1 nodes and using quantification rules given in reference [**12**]. Consider the BDD representing mission failure in phase 2 given in Fig. 3. This has two paths to its single terminal 1 node, which are disjoint and

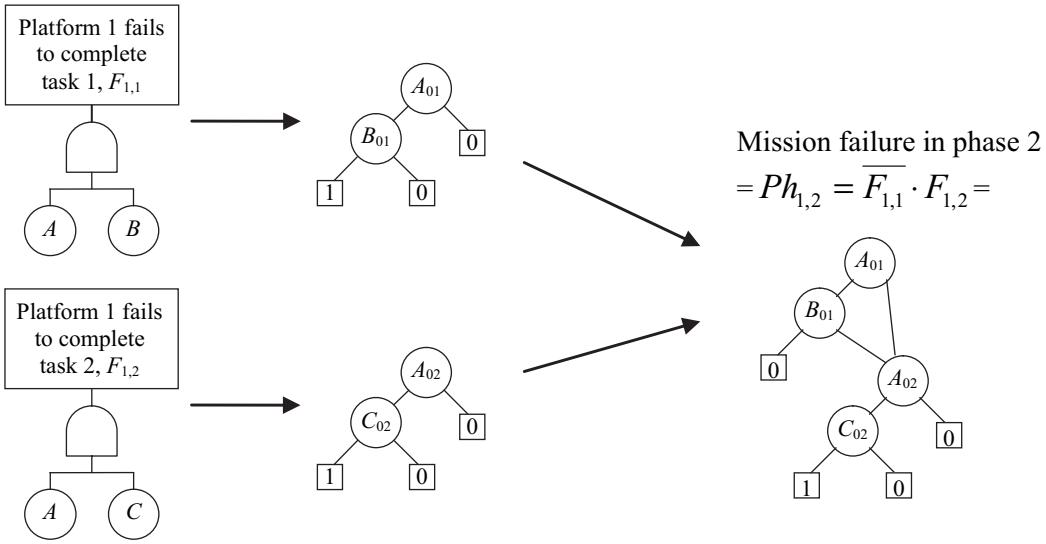


Fig. 3 A BDD representing failure in phase 2 of a simple mission

represent the logical expression for the failure of platform 1 in phase 2. The paths are shown in the following equation, where each of the path expressions is also simplified using the path rules from reference [**12**]

$$Ph_{1,2} = \begin{bmatrix} A_{01} \cdot B_{1\infty} \cdot A_{02} \cdot C_{02} \\ +A_{1\infty} \cdot A_{02} \cdot C_{02} \end{bmatrix} = \begin{bmatrix} A_{01} \cdot B_{1\infty} \cdot C_{02} \\ +A_{12} \cdot C_{02} \end{bmatrix} \tag{5}$$

If a variable is traversed on its 1-branch (left) then that variable is true and is included in the path as such. If the variable is traversed on its 0-branch (right) that variable is false and is included in the path as such. Note that, for the variables considered here

$$\overline{x_{0i}} = x_{i\infty} \tag{6}$$

since if the variable does not occur in the first i phases of the mission it must occur at some time afterwards. Thus this representation is used when traversing the 0-branch of a variable.

The paths are simplified using a process that is described in detail in reference [**12**]. In order to illustrate what happens in this example consider each path in turn and look for repeated variables on the paths. In the first path A appears twice as A_{01} and A_{02}. These variables represent respectively the occurrence of A in the first phase and the occurrence of A in the first phase or the second phase. For each of these to be true A must fail in the first phase. Thus the variables representing A reduce to A_{01}. In general

$$x_{i_1 j_1} \cdot x_{i_2 j_2} = x_{[\max(i_1, i_2)][\min(j_1, j_2)]} \tag{7}$$

If, for a general variable x_{ij}, $i \geqslant j$ then $x_{ij} = 0$. Further details can be found in reference [**12**]. Once the paths have been determined and their logic simplified the probability of occurrence can be determined for each of them. The path probabilities are added to give the total probability. When doing this the probability for each variable is determined using

$$P(x_{ij}) = \int_{t_i}^{t_j} f_x(t)\,\mathrm{d}t \tag{8}$$

where $f_x(t)$ is the failure probability density function for the component represented by variable x.

3 MULTIPLATFORM PHASED MISSION METHODOLOGY

Presented above was an overview of a method for quantifying the probability of phase failure for single platforms performing a phased mission. This section presents a methodology for analysing the failure probability of multiplatform phased missions. Firstly, a technique for calculating phase and mission failure probabilities before a mission begins, i.e. initial failure probabilities, is presented. Next, a technique for calculating updated failure probabilities, calculated once the mission is in progress, is presented. These updated failure probabilities are calculated taking into account any information that is currently known about the mission progress, which could, for example, come from a diagnostics tool. Different cases are considered for this information, and then the time at which this information becomes available is considered.

3.1 Definitions

Consider a multiplatform phased mission being conducted by n platforms, each of which is performing its own phased mission, part of which is a task or tasks that contribute to achieving the overall mission objective. Assume that the platforms are considered to be in a fully working state when they begin their own phased mission (which may or may not begin as the entire mission begins) and that all of the platforms are non-repairable over the mission duration. Given that each platform performs a number of phases as part of its own phased mission and that the phases of different platforms do not necessarily begin and end simultaneously, it is possible to identify a number of distinct mission phases, m. The start and end times of these mission phases will coincide with each distinct point in time at which platform phases begin and end. Thus the minimum number of mission phases, m, is the number of phases carried out by the platform which performs the greatest number of phases. The maximum value will be the sum of the number of phases in all of the individual platform missions.

As already described for single-platform phased missions, a platform will successfully complete its phased mission only if it completes every single phase of that mission successfully. When a number of platforms work together to achieve a common objective only certain phases that are performed by the individual platforms will contribute to the overall success of the mission. For example, if reconnaissance performed by an unmanned aerial vehicle (UAV) is crucial to the success of a multiplatform phased mission then it might not be imperative that the UAV is successfully recovered after its reconnaissance phase has been performed. It could, however, also be the case that each of the platforms involved in the multiplatform phased mission has to successfully complete all of its own phases in order for the mission to be successful. For example, the return to base of the UAV might be considered crucial to the overall success of the mission. In general, the importance of each of the individual platforms

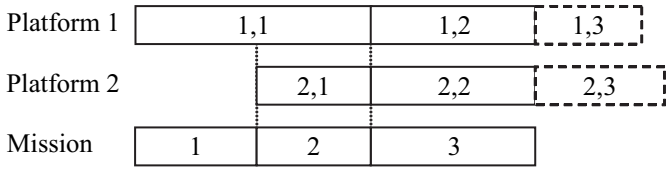


Fig. 4 A simple multiplatform phased mission representation

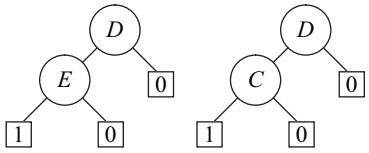


Fig. 5 The BDDs for the failure conditions of platform 2 in its first phase (left) and second phase (right)

completing their own phased mission as part of the multiplatform mission will depend upon the mission being performed.

Each of the platforms taking part in the multiplatform phased mission will have certain functions and subsystems that are unique to them. However, there could also be functions and subsystems that are used by more than one platform and hence introduce dependencies between the platforms. This should come as no surprise since the platforms are not working independently to achieve the overall mission objective. A dependency between platforms could, for example, be introduced by a shared communication system.

A simple example of a multiplatform phased mission is introduced in Fig. 4. Here, two platforms, 1 and 2, are each required to perform two tasks, which contribute to the overall mission. The first task is to reach a meeting point; the second is to each perform a task at that meeting point. Each platform then goes on to perform a third phase, which does not contribute to the overall mission. If the phased missions were to be analysed for the individual platforms these phases would be taken account of. However, they will not be taken account of when determining the successful achievement of the overall mission objective. Due to the platform capabilities and initial locations the journey to the meeting point takes longer for platform 1 than for platform 2. Thus, as can be seen from the diagram, there are distinct phases in achieving the overall mission objective:

1. Platform 1 begins the journey to the meeting point.
2. Platform 1 continues towards the meeting point and platform 2 starts its journey to the meeting point.
3. Both platforms perform the tasks in their second phases together.

The phase failure logic expressions for platform 1 are those given in the BDDs included in the simple example in Fig. 3, and the phase failure logic expression BDDs for platform 2 are shown in Fig. 5. Note that the failure logic expression BDDs are not given for the third platform phases since these do not contribute to the overall mission objective that is being analysed here. It should also be noted that the failure of platform 2 in its second phase depends on the occurrence of *C*. This represents a dependency between the platforms, since the failure of platform 1 in its own second phase also depends on *C*. In reality, such a dependency could come from a shared communication link, for example.

Initial and updated failure probabilities can be calculated for the individual platforms using the methods discussed in section 2.1. For the whole mission, the methods to be used are detailed in the following sections. The assumptions specific to the case of multiple platforms are as follows:

1. In order for the mission to fail in any phase it must have successfully completed all previous mission phases.
2. Each platform in a multiplatform phased mission is considered to be fully functional at the start of its own mission. The only exception to this is when failure events are shared by platforms (i.e. dependencies exist between platforms). In this case it is assumed that the event may occur from the time when the first platform that is dependent upon that event starts its own mission.
3. Mission failure in any particular mission phase is defined to occur when at least one of the individual platforms performing in that phase fails to complete that phase. (Any redundancy whereby other platforms could perform the task required of the failing platform is assumed to entail the employment of a different mission configuration.)

Note that in reality multiplatform phased missions are likely to be very complex, featuring a greater number of platforms, each of which is required to perform more phases that contribute to the overall mission. The example given is purely to demonstrate the concepts of multiplatform phased mission analysis.

3.2 Calculating initial probabilities

Let F_i represent the logical expression for the failure conditions for the entire mission being met in mission phase i and let Ph_i represent the logical expression for mission failure in mission phase i. Then

$$Ph_i = \overline{F_1} \cdot \overline{F_2} \cdot \overline{F_3} \cdot \cdots \cdot \overline{F_{i-1}} \cdot F_i \tag{9}$$

This is in accordance with the assumption that in order for the mission to fail in mission phase i all of the mission phases from 1 to $i-1$ must have been completed successfully. For the mission failure conditions to be met in mission phase i at least one of the platforms that are taking part in the mission must fail in that phase. Therefore

$$F_i = F_{1,i} + F_{2,i} + \cdots + F_{n,i} \tag{10}$$

where $+$ represents the Boolean OR operator and n is the number of platforms involved in the mission. Substituting equation (10) into equation (9) gives the logical expression for mission failure in phase i in terms of the logical expressions for the failure conditions of the individual platforms being met

$$\begin{aligned} Ph_i = {} & \overline{F_{1,1}} \cdot \overline{F_{1,2}} \cdot \overline{F_{1,3}} \cdot \cdots \cdot \overline{F_{1,i-1}} \cdot \\ & \cdot \overline{F_{2,1}} \cdot \overline{F_{2,2}} \cdot \overline{F_{2,3}} \cdot \cdots \cdot \overline{F_{2,i-1}} \cdot \\ & \qquad \vdots \\ & \cdot \overline{F_{n,1}} \cdot \overline{F_{n,2}} \cdot \overline{F_{n,3}} \cdot \cdots \cdot \overline{F_{n,i-1}} \cdot \\ & \cdot \left(F_{1,i} + F_{2,i} + \cdots + F_{n,i}\right) \end{aligned} \tag{11}$$

Note that since it is not necessarily the case that all platforms are active throughout the entire mission, it could be that some of these $F_{p,i}$'s could be set to zero. For example, if platform 1 only began its own phased mission in mission phase 4 of the overall mission, then $F_{1,1}$, $F_{1,2}$, and $F_{1,3}$ would be set to zero. A similar case holds if a platform finishes its mission when there are still further mission phases to be conducted. For example, if platform 1 finishes its own phased mission in the overall mission phase 10 and equation (11) is constructed for $i = 12$, then $F_{1,11}$ and $F_{1,12}$ must be set to zero. This is, of course, also the case when later platform phases do not contribute to mission success.

Finding the mission phase failure probabilities, Q_i, is carried out using the logical expressions formed using equation (11). These phase failure probabilities are then added to give the total failure probability for the entire mission, Q_{MISS}

$$Q_{\text{MISS}} = \sum_{i=1}^{m} Q_i \tag{12}$$

For the two-platform missions shown in Fig. 4 the BDDs representing the logical expressions for the mission phase failure probabilities given in equation (11) can be constructed as shown in Fig. 6. The parts of the BDD representing the logic for the different platform mission phase failure conditions are marked with dashed lines. Note how, for the BDDs representing platform 2, i.e. $F_{2,2}$ and $F_{2,3}$, the allocated time intervals for the variables start at 1 rather than 0 (as is the case for platform 1), since platform 2 starts its own phased mission as mission phase 1 ends, not at the start of the mission. Note also that, for the same reason, $F_{2,1}$ is set to zero since platform 2 is not yet active in mission phase 1. It should also be noted that, when considering $F_{2,3}$, the variable C is allocated the time interval (0, 3) rather than (1, 3), in accordance with the assumption that shared events are considered able to fail from the start of the platform starting first of all with the platforms sharing the variable.

However, it is better to construct the BDDs representing mission phase failure by taking into account the platform phases wherever possible, since this will reduce the size of the phase failure BDDs. To illustrate this consider the BDD for Ph_3 in Fig. 6. Since platform 1 is performing its own first phase over mission phases 1 and 2, this can be represented by a single BDD, which is equivalent to

$$\overline{F_{1,1}} \cdot \overline{F_{1,2}} \tag{13}$$

Figure 7 shows the result of this. Performing this simple process wherever possible will help to minimize the size of the BDDs representing Ph_i. For example, the BDD representing Ph_3 in Fig. 7 has two fewer nodes than that in Fig. 6 (8 as opposed to 10) and half as many 1-terminating paths (12 as opposed to 24). This will have a positive effect on the quantification time.

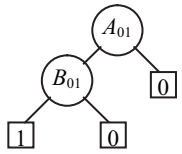
The paths of the BDDs can then be determined and simplified. The resultant logic for the mission failure in phases 1, 2, and 3 is given in the following equations respectively, showing that these expressions can be quantified as demonstrated earlier to give the mission phase failure probabilities

$$Ph_1 = A_{01} \cdot B_{01} \tag{14}$$

$$\begin{aligned} Ph_2 &= \begin{bmatrix} A_{01} \cdot B_{1\infty} \cdot A_{02} \cdot B_{02} \\ + A_{01} \cdot B_{1\infty} \cdot A_{02} \cdot B_{2\infty} \cdot D_{12} \cdot E_{12} \\ + A_{01} \cdot B_{1\infty} \cdot A_{2\infty} \cdot D_{12} \cdot E_{12} \\ + A_{1\infty} \cdot A_{02} \cdot B_{02} \\ + A_{1\infty} \cdot A_{02} \cdot B_{2\infty} \cdot D_{12} \cdot E_{12} \\ + A_{1\infty} \cdot A_{2\infty} \cdot D_{12} \cdot E_{12} \end{bmatrix} \\ &= \begin{bmatrix} A_{01} \cdot B_{12} \\ + A_{01} \cdot B_{2\infty} \cdot D_{12} \cdot E_{12} \\ + 0 \\ + A_{12} \cdot B_{02} \\ + A_{12} \cdot B_{2\infty} \cdot D_{12} \cdot E_{12} \\ + A_{2\infty} \cdot D_{12} \cdot E_{12} \end{bmatrix} \end{aligned} \tag{15}$$

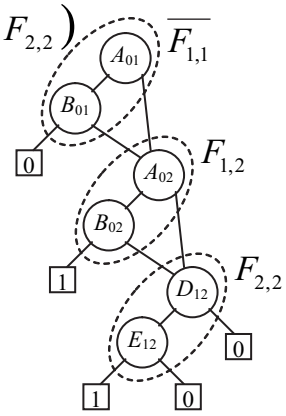
Mission failure in phase 1:

$$Ph_1 = F_{1,1} + F_{2,1} = F_{1,1}$$



Mission failure in phase 2:

$$Ph_2 = \overline{F_{1,1}} \cdot \overline{F_{2,1}} \cdot \left(F_{1,2} + F_{2,2}\right) = \overline{F_{1,1}} \cdot \left(F_{1,2} + F_{2,2}\right)$$



Mission failure in phase 3:

$$Ph_3 = \overline{F_{1,1}} \cdot \overline{F_{1,2}} \cdot \overline{F_{2,1}} \cdot \overline{F_{2,2}} \cdot \left(F_{1,3} + F_{2,3}\right) = \overline{F_{1,1}} \cdot \overline{F_{1,2}} \cdot \overline{F_{2,2}} \cdot \left(F_{1,3} + F_{2,3}\right)$$

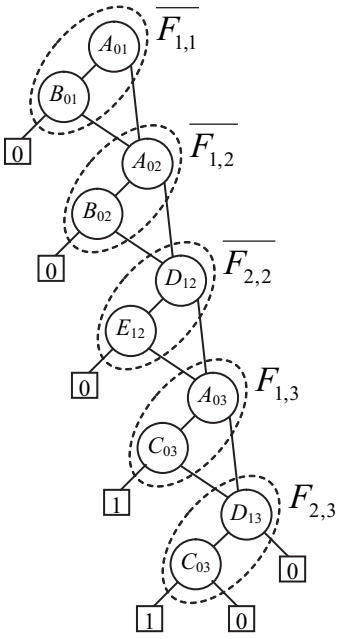


Fig. 6 The BDDs representing mission phase failure for each of the three mission phases of the example mission

$$Ph_3 = \begin{bmatrix} A_{02} \cdot B_{2\infty} \cdot D_{12} \cdot E_{2\infty} \cdot A_{03} \cdot C_{03} \\ + A_{02} \cdot B_{2\infty} \cdot D_{12} \cdot E_{2\infty} \cdot A_{03} \cdot C_{3\infty} \cdot D_{13} \cdot C_{03} \\ + A_{02} \cdot B_{2\infty} \cdot D_{12} \cdot E_{2\infty} \cdot A_{3\infty} \cdot D_{13} \cdot C_{03} \\ + A_{02} \cdot B_{2\infty} \cdot D_{2\infty} \cdot A_{03} \cdot C_{03} \\ + A_{02} \cdot B_{2\infty} \cdot D_{2\infty} \cdot A_{03} \cdot C_{3\infty} \cdot D_{13} \cdot C_{03} \\ + A_{02} \cdot B_{2\infty} \cdot D_{2\infty} \cdot A_{3\infty} \cdot D_{13} \cdot C_{03} \\ + A_{2\infty} \cdot D_{12} \cdot E_{2\infty} \cdot A_{03} \cdot C_{03} \\ + A_{2\infty} \cdot D_{12} \cdot E_{2\infty} \cdot A_{03} \cdot C_{3\infty} \cdot D_{13} \cdot C_{03} \\ + A_{2\infty} \cdot D_{12} \cdot E_{2\infty} \cdot A_{3\infty} \cdot D_{13} \cdot C_{03} \\ + A_{2\infty} \cdot D_{2\infty} \cdot A_{03} \cdot C_{03} \\ + A_{2\infty} \cdot D_{2\infty} \cdot A_{03} \cdot C_{3\infty} \cdot D_{13} \cdot C_{03} \\ + A_{2\infty} \cdot D_{2\infty} \cdot A_{3\infty} \cdot D_{13} \cdot C_{03} \end{bmatrix} = \begin{bmatrix} A_{02} \cdot B_{2\infty} \cdot C_{03} \cdot D_{12} \cdot E_{2\infty} \\ + 0 \\ + 0 \\ + A_{02} \cdot B_{2\infty} \cdot C_{03} \cdot D_{2\infty} \\ + 0 \\ + 0 \\ + A_{23} \cdot C_{03} \cdot D_{12} \cdot E_{2\infty} \\ + 0 \\ + A_{3\infty} \cdot C_{03} \cdot D_{12} \cdot E_{2\infty} \\ + A_{23} \cdot C_{03} \cdot D_{2\infty} \\ + 0 \\ + A_{3\infty} \cdot C_{03} \cdot D_{23} \end{bmatrix} \tag{16}$$

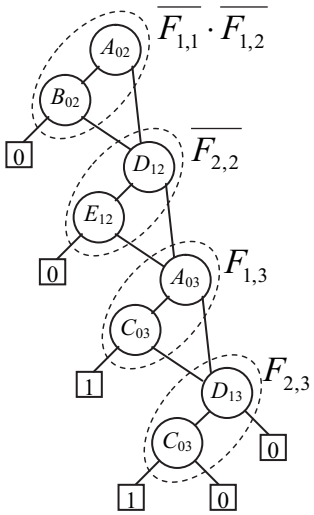


Fig. 7 BDD representing mission phase 3 failure when calculating initial mission failure probabilities

3.3 Calculating updated probabilities

Once a mission is underway and several phases are successfully completed, the probability of mission failure for remaining mission phases can be calculated in order to give an updated measure of the mission failure probability. In such cases it is necessary to take into account that a certain portion of the mission will have been successfully completed in order to reach the point at which the updated probabilities are required. Bayes' theorem of conditional probability states that the probability of A occurring given that an event B has occurred is given as follows

$$P(A|B) = \frac{P(A \cdot B)}{P(B)} \tag{17}$$

This expression can be used to find the probability of failure in a certain mission phase given that a number of mission phases have been successfully completed. The probability of mission failure in mission phase j given the successful completion of k phases is thus given by

$$\begin{aligned} Q_{j|\overline{k}} &= \frac{P\left(Ph_j \cdot \overline{Ph_1} \cdot \overline{Ph_2} \cdot \cdots \cdot \overline{Ph_k}\right)}{P\left(\overline{Ph_1} \cdot \overline{Ph_2} \cdot \cdots \cdot \overline{Ph_k}\right)} \\ &= \frac{P(Ph_j)}{1 - P(Ph_1 + Ph_2 + \cdots + Ph_k)} \end{aligned} \tag{18}$$

Since the mission phases are mutually exclusive this is equivalent to

$$Q_{j|\overline{k}} = \frac{Q_j}{1 - \sum_{i=1}^{k} Q_i} \tag{19}$$

These mission phase failure probabilities given k successfully completed mission phases can now be added to give the total probability of mission failure given k successfully completed mission phases

$$Q_{\mathrm{MISS}|\overline{k}} = \sum_{j=k+1}^{m} Q_{j|\overline{k}} \tag{20}$$

For the two-platform example multiplatform phased missions the failure probabilities obtained when quantifying the BDDs given in Fig. 6 would be substituted into equation (19) in order to calculate these updated probabilities.

3.4 Including diagnostics information

The updated failure probabilities given in section 3.3 are calculated taking into account the mission phases completed. However, no account is taken of the change in status of any parts of the platforms that are performing the multiplatform phased mission. It is likely that, as the mission progresses, information will become available about certain parts of the systems involved. For example, a diagnostics device might report that a component of a system failed at a specified time or that a function was known to be successful at a certain time. It is clearly important to take account of this kind of information.

3.4.1 Taking account of subsystem status

Assume that there is a subsystem, S, which is required by one or more of the platforms taking part in the multiplatform phased mission, and that the subsystem will be considered to have failed at the point that the diagnostics device reports the failure of S. It is possible that S could be used to perform different functions in different mission phases. However, if S is reported to have failed in a particular configuration, then it is assumed that S has not failed in that configuration at an earlier time. Let the logical expression for subsystem S failure in a particular configuration in mission phase i be represented by $F_{S,i}$. Since it has not failed in that configuration at an earlier time that failure configuration must not have occurred earlier in the mission, i.e. in the previous $i-1$ phases. Therefore the logical expression for the failure of subsystem S in phase i is given by considering its successful operation through previous phases and failure in mission phase i as follows

$$Ph_{S,i} = \overline{F_{S,1}} \cdot \overline{F_{S,2}} \cdot \overline{F_{S,3}} \cdot \cdots \cdot \overline{F_{S,i-1}} \cdot F_{S,i} \tag{21}$$

Assume that subsystem S is known to have failed after the completion of r mission phases and that the mission then continues. After k ($k \geqslant r$) mission phases have been successfully completed, Bayes' theorem (equation (17)) can be used to give an expression for the failure probability of mission phase j given the

successful completion of k mission phases and subsystem S failure after r mission phases

$$\begin{aligned} Q_{j|\overline{k}.S,r} &= P\left(\text{the mission fails in phase } j \;\middle|\; \begin{array}{c}\text{phases 1 to } k \text{ were successfully completed}\\ \text{AND}\\ \text{subsystem } S \text{ failed in phase } r\end{array}\right) \\ &= P\left(Ph_j \middle| \overline{Ph_1}\cdot\overline{Ph_2}\cdot\cdots\cdot\overline{Ph_k}\cdot Ph_{S,r}\right) \\ &= \frac{P\left(Ph_j\cdot\overline{Ph_1}\cdot\overline{Ph_2}\cdot\cdots\cdot\overline{Ph_k}\cdot Ph_{S,r}\right)}{P\left(\overline{Ph_1}\cdot\overline{Ph_2}\cdot\cdots\cdot\overline{Ph_k}\cdot Ph_{S,r}\right)} \end{aligned} \tag{22}$$

This does not simplify to give an equation of the form of that in equation (20), since the terms representing mission phase failure in phases are not mutually exclusive from the terms representing subsystem failure in mission phases. However, equation (22) can be reduced to

$$Q_{j|\overline{k}.S,r} = \frac{P\left(Ph_j\cdot Ph_{S,r}\right)}{P\left(\overline{F_1}\cdot\overline{F_2}\cdot\cdots\cdot\overline{F_k}\cdot Ph_{S,r}\right)} \tag{23}$$

Using equations (10), (11), and (21), the logical expressions for which failure probabilities must be calculated can be converted to a form wherein all single terms are logical expressions for the failure conditions of the platforms being met in each of the mission phases, $F_{p,i}$, and for the subsystem S failure conditions being met in each of the mission phases, $F_{S,i}$. The BDDs representing these expressions can then be constructed using the BDDs for $F_{p,i}$ and $F_{S,i}$, as shown in section 2.2, and quantified to give the updated phase failure probabilities. In order to find the total failure probability given that k mission phases have been successfully completed and subsystem S failed after r mission phases, the phase failure probabilities are added over the remaining phases

$$Q_{\text{MISS}|\overline{k}.S,r} = \sum_{j=k+1}^{m} Q_{j|\overline{k}.S,r} \tag{24}$$

If the subsystem S was instead known to have worked for a certain number of phases then this information could be included in a similar way in order to calculate updated failure probabilities. In this case the probability of failure in mission phase j, given that k phases have been successfully completed and subsystem S is known to have functioned successfully for r phases, is given by

$$Q_{j|\overline{k}.\overline{S,r}} = \frac{P\left(Ph_j\cdot\overline{F_{S,1}}\cdot\overline{F_{S,2}}\cdot\cdots\cdot\overline{F_{S,r}}\right)}{P\left(\overline{F_1}\cdot\overline{F_2}\cdot\cdots\cdot\overline{F_k}\cdot\overline{F_{S,1}}\cdot\overline{F_{S,2}}\cdot\cdots\cdot\overline{F_{S,r}}\right)} \tag{25}$$

The expressions given in equations (23) and (25) are constructed using the platform phase failure logic BDDs and the subsystem phase failure logic BDDs in a similar way to that demonstrated when calculating the initial failure probabilities.

3.4.2 *Taking account of the latest component failure data*

When a subsystem S failure occurs, as described in the previous section, a diagnostics system could take into account the fault symptoms reported and use these to provide information as to which system components caused the failure. Even if it is not possible to determine exactly which components caused S to fail, evidence of the fault symptoms can be utilized to deduce the failure probability of those components with the potential to have failed. There will be two possibilities for such components:

1. A new failure probability density function is provided by the diagnostics system, giving the probability that the component will fail after the time of the diagnosis.
2. The diagnostics system provides a certain probability of the component being in a failed state at the time of the diagnosis.

This information is then used along with the fact that the subsystem has failed (as shown in the previous section) in order to quantify the failure probability over the remainder of the mission. This is done by taking into account new quantification rules for components when calculating path probabilities in the mission phase BDDs. Both of these cases are now discussed below.

Case 1

If the information is provided from a diagnostics tool at time t_d then there will be a new failure probability density function for component x after time t_d given by $f_{x,d}(t)$. In this case path simplification takes place as in section 2.3. However, when finding the probability of path occurrence, equation (8) is no longer used to calculate the probabilities for variables on the path whose failure probability density function has been updated. Instead, the following equation is

used, where, for any probability before t_d, the original failure density function $f_x(t)$ is used, and after t_d the new density function $f_{x,d}(t)$ is used

$$P(x_{ij}) = \begin{cases} \int_{t_i}^{t_j} f_x(t)\,dt, & t_i < t_d, \quad t_j \leqslant t_d \\ \int_{t_i}^{t_d} f_x(t)\,dt + \int_{t_d}^{t_j} f_{x,d}(t)dt, & t_i < t_d, \quad t_j > t_d \\ \int_{t_i}^{t_j} f_{x,d}(t)\,dt, & t_i \geqslant t_d \end{cases} \tag{26}$$

Case 2

For Case 2, the output of the diagnostics tool for the components is a probability $q_{x,d}$, which is independent of time. Now, owing to the evidence of the subsystem failure, the component is believed to have failed with a constant probability. Thus, at any time during the mission, the probability that the component has failed is $q_{x,d}$. Also, after the time of the diagnosis event, t_d, the component is now considered to remain in the same state that it was in immediately before t_d, since no new failure distribution is available unless new information is reported by the diagnostics tool. The quantification process takes this information into account using the following equation instead of equation (8) for relevant components

$$P(x_{ij}) = \begin{cases} q_{x,d}, & t_j \leqslant t_d \\ 1 - q_{x,d}, & t_i \leqslant t_d, \quad t_j = \infty \\ 0, & \text{otherwise} \end{cases} \tag{27}$$

Quantification. If updated component failure probabilities are provided by a diagnostics system then they must be taken into account during the quantification. If the component failure data supplied is in the form of failure probability distributions (Case 1), then once the updated phase failure probabilities are calculated for the remaining phases of the mission these updated failure probabilities may be added to give the total mission failure probability as in equation (20). However, if the component failure data supplied are in terms of a probability of the component being in a failed state (Case 2), then the phase failure probabilities cannot be added to give the total mission failure probability since the component failure probability contribution to the specific mission phases cannot be ascertained. In this case logical expressions must be constructed and analysed separately for each mission phase and the mission as a whole.

3.4.3 *Taking account of the exact time of diagnostics information*

As considered so far, the updated failure probabilities are calculated assuming that the time at which the diagnostics information becomes available is a point between mission phases, as one ends and another starts. It is likely that the phase and mission failure probabilities will be required from some time during a mission phase. If diagnostics information comes to light during a mission phase it is possible to calculate the failure probability for the remainder of the phase given the information known up to this time. For situations such as this, the mission can be split at the point where the information becomes available, and the updated mission phase failure probabilities are calculated using the expressions derived earlier, given that the mission was successful for previous mission phases.

4 EXAMPLE ANALYSIS

In order to illustrate the BDD analysis techniques for multiplatform phased missions consider again the two-platform phased mission described earlier. Assume that there is a subsystem S in this mission whose failure will be reported by a diagnostics tool and that S is made up simply of component A. This is a simplistic example for demonstration purposes here and in reality it is unlikely that a subsystem will be made up of a single component. However, the methodology works in just the same way.

Consider that the mission is in progress and that a fault is diagnosed in mission phase 2, as shown in Fig. 8, where the time of the diagnosis is illustrated by the thick dashed line. New updated mission failure probabilities are required at this point in order to predict the failure probability for the remainder of the mission. As described earlier, when taking into account the exact time of diagnostics information, the mission phase in which the diagnosis is made is split. Thus, for this example, mission phase 2 is split, giving new mission phase x, which is the completed part of mission phase 2, and y, which is the part of mission phase 2 still to be completed. Mission phases x and y are also shown in Fig. 8.

Now the updated mission failure probabilities can be calculated using equation (23), given that mission phases 1 and x have been successfully completed and

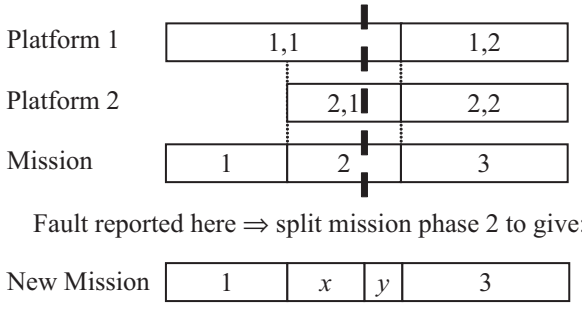


Fig. 8 Splitting a mission phase at the time of a diagnostics report

that subsystem S worked successfully through phases 1 but failed in phase x. Using equation (23) gives

$$\begin{aligned} Q_{y|\overline{x}.S,x} &= P\left(\text{the mission fails in phase } y \;\middle|\; \begin{array}{c}\text{phases 1 and } x \text{ were successfully completed}\\ \text{AND}\\ \text{subsystem } S \text{ failed in phase } x\end{array}\right) \\ &= \frac{P\left(Ph_y \cdot Ph_{S,x}\right)}{P\left(\overline{F_1} \cdot \overline{F_x} \cdot Ph_{S,x}\right)} \\ &= \frac{P\left(\overline{F_{1,1}} \cdot \overline{F_{1,x}} \cdot \overline{F_{2,x}} \cdot \left(F_{1,y} + F_{2,y}\right) \cdot \overline{F_{S,1}} \cdot F_{S,x}\right)}{P\left(\overline{F_{1,1}} \cdot \overline{F_{1,x}} \cdot \overline{F_{2,x}} \cdot \overline{F_{S,1}} \cdot F_{S,x}\right)} \end{aligned} \tag{28}$$

and

$$\begin{aligned} Q_{3|\overline{x}.S,x} &= P\left(\text{the mission fails in phase 3} \;\middle|\; \begin{array}{c}\text{phases 1 and } x \text{ were successfully completed}\\ \text{AND}\\ \text{subsystem } S \text{ failed in phase } x\end{array}\right) \\ &= \frac{P\left(Ph_3 \cdot Ph_{S,x}\right)}{P\left(\overline{F_1} \cdot \overline{F_x} \cdot Ph_{S,x}\right)} \\ &= \frac{P\left(\overline{F_{1,1}} \cdot \overline{F_{1,x}} \cdot \overline{F_{1,y}} \cdot \overline{F_{2,x}} \cdot \overline{F_{2,y}} \cdot \left(F_{1,3} + F_{2,3}\right) \cdot \overline{F_{S,1}} \cdot F_{S,x}\right)}{P\left(\overline{F_{1,1}} \cdot \overline{F_{1,x}} \cdot \overline{F_{2,x}} \cdot \overline{F_{S,1}} \cdot F_{S,x}\right)} \end{aligned} \tag{29}$$

Equations (28) and (29) share a denominator, the BDD for which is shown in Fig. 9. Note that when constructing this BDD its size is minimized by taking account of the mission phases that form complete or partial platform phases. Phase 1 for platform 1 actually takes place over mission phases 1, x, and y, and therefore this fact is taken into account when constructing the BDD. The portions of the BDD in Fig. 9 representing platform 1 failure in mission phase 1 and mission phase x are hence denoted by a single BDD.

Figure 10 shows the BDDs for the logical expressions in the numerators of equations (28) and (29). Note again how the size of the BDDs has been kept to a minimum by taking account of consecutive mission phases that make up part of a single-platform phase when considering the logical expressions for mission phase failure and combining BDDs accordingly. These BDDs are quantified as before.

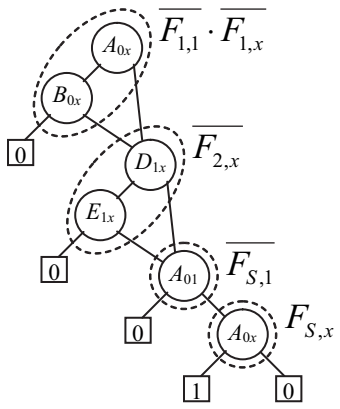


Fig. 9 The BDD representing the denominator of equations (28) and (29)

5 CONCLUSIONS

A methodology for quantifying phase and mission failure probabilities for multiplatform phased mission systems has been presented. It has characteristics that make it suitable for use in a decision-making strategy for missions involving multiplatform collaboration. The methodology allows:

1. Rapid connection of BDDs representing platform failure in mission phases, to give BDDs representing mission failure in those phases as soon as new or alternative mission configurations become known. This means that quantification may begin relatively soon after a mission configuration is proposed.
2. The calculation of initial failure probabilities before a mission begins and updated failure probabilities once a mission is in progress.
3. The use of initial mission phase failure probabilities to calculate the updated mission phase failure probabilities when no further information is known about the system.
4. The calculation of updated mission failure probabilities when diagnostics information such

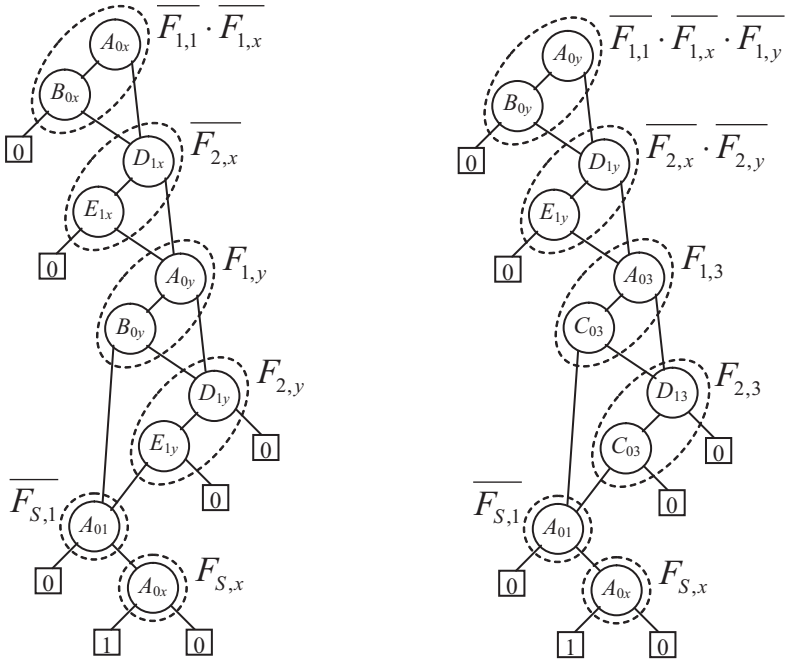


Fig. 10 The BDDs representing the numerators of equations (28) (left) and (29) (right)

as a fault or other change in the mission environment becomes known. The cases of subsystem success and failure are covered.

5. The use of further diagnostics information regarding updated component failure data when calculating updated failure probabilities.
6. The exact time of diagnostics events to be considered when calculating updated failure probabilities.

Since BDDs offer a rapid reliability evaluation capability they form the basis of the methodology. This provides the greatest opportunity to develop a method capable of providing real-time decisions for fast decision making.

ACKNOWLEDGEMENTS

The authors would like to thank EPSRC and BAE Systems for the financial support for this research which contributed to the NECTISE (Network Enabled Capability Through Innovative Systems Engineering) project.

REFERENCES

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

2 **Vyzaite, G., Dunnett, S.,** and **Andrews, J. D.** Cause consequence analysis of non-repairable phased missions. *Reliability Engineering and System Safety*, 2006, **91**, 398–406.

3 **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**(E1), 83–91.

4 **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.

5 **Smotherman, M.** and **Zemoudeh, K.** A non-homogeneous Markov model for phased-mission reliability-analysis. *IEEE Trans. Reliability*, 1989, **38**(5), 585–590.

6 **Altschul, R. E.** and **Nagel, P. M.** The efficient simulation of phased fault trees. In Proceedings of the Annual Reliability and Maintainability Symposium, Philadelphia, 1987, pp. 292–296.

7 **Mura, I.** and **Bondavalli, A.** Markov regenerative stochastic Petri nets to model and evaluate phased mission systems dependability. *IEEE Trans. Computers*, 2001, **50**(12), 1337–1351.

8 **Ou, Y., Meshkat, L.,** and **Dugan, J. B.** Multi-phase reliability analysis for dynamic and static phases. In Proceedings of the Annual Reliability and Maintainability Symposium, Seattle, 2002, pp. 404–410.

9 **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, 2008, pp. 8–14.

10 **Andrews, J. D.** A ternary decision diagram method to calculate the component contribution to the failure of systems undergoing phased missions. *Proc. IMechE, Part O: J. Risk and Reliability*, 2008, **222**(O2), 173–187.

11 **Rauzy, A.** New algorithms for fault tree analysis. *Reliability Engineering and System Safety*, 1993, **40**, 203–211.

12 **Prescott, D. R., Remenyte-Prescott, R., Reed, S., Andrews, J. D.,** and **Downes, C. G.** A reliability analysis method using BDDs in phased mission planning. *Proc. IMechE, Part O: J. Risk and Reliability*, 2008 (submitted).