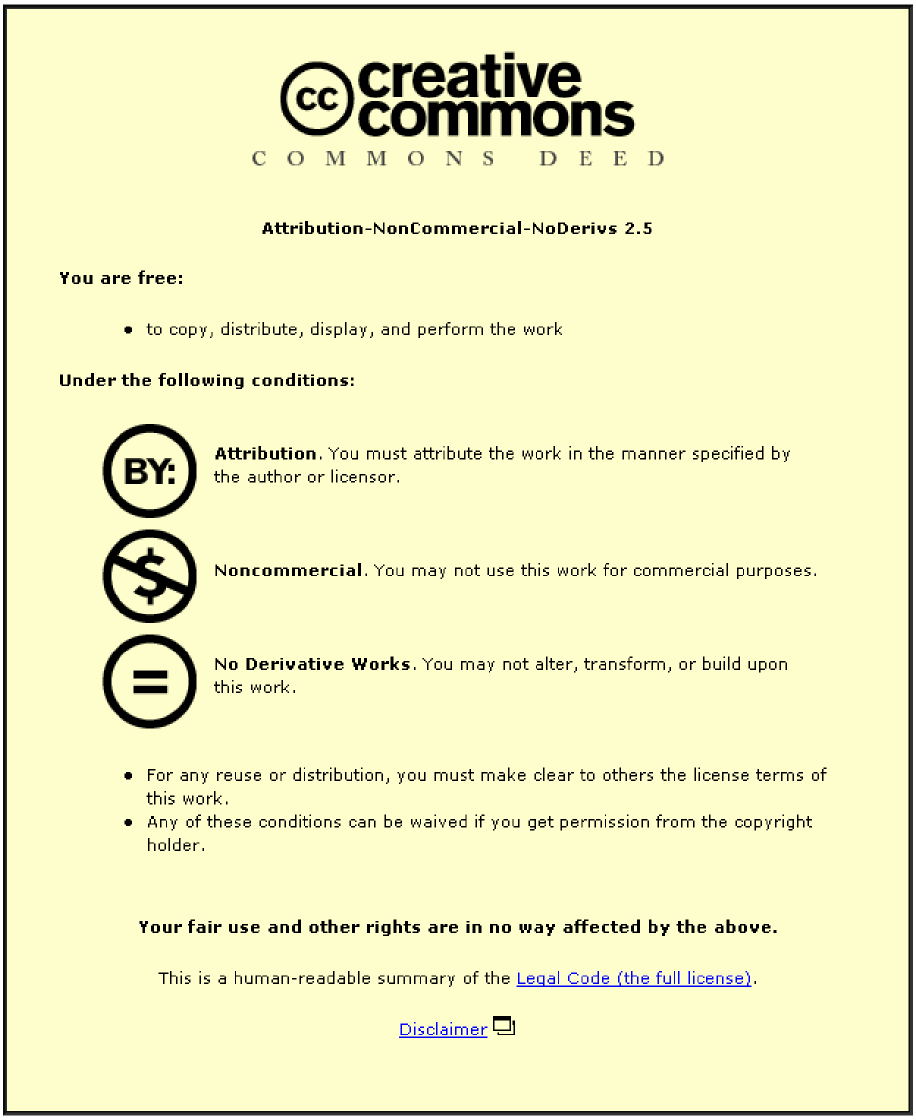
Phased mission modelling using fault tree analysis

La Band, Rachel A., and J.D. Andrews. 2008. “Phased Mission Modelling Using Fault Tree Analysis”. figshare. https://hdl.handle.net/2134/3732.



Phased Mission Modelling using Fault Tree Analysis

R A La Band and **J D Andrews**
Department of Systems Engineering, Loughborough University, Leicestershire, UK

Abstract: Many types of system operate for missions which are made up of several phases. For the complete mission to be a success, the system must operate successfully during each of the phases. Examples of such systems include an aircraft flight, and also many military operations for both aircraft and ships. An aircraft mission could be considered as the following phases: taxiing to the runway, take-off, climbing to the correct altitude, cruising, descending, landing and taxiing back to the terminal. Component failures can occur at any point during the mission but their condition may only be critical for one particular phase. As such it may be that the transition from one phase to another is the critical event leading to mission failure, the component failures resulting in the system failure may have occurred during some previous phase.

This paper describes a means of analysing the reliability of non-repairable systems which undergo phased missions. Fault Tree Analysis has been used as the method to assess the system performance. The results of the analysis are the system failure modes in each phase (minimal cut sets), the failure probability in each phase and the total mission unreliability. To increase the efficiency of the analysis the fault trees constructed to represent the system failure logic are analysed using a modularisation method. Binary Decision Diagrams (BDD's) are then employed to quantify the likelihood of failure in each phase.

Keywords: phased missions, fault tree analysis, binary decision diagrams

1 Introduction

If the success of a mission is reliant upon a set of objectives operating over different time intervals, it may be referred to as a *Phased Mission*. During the execution of the phases in a mission, the system is altered such that the logic model, system configuration, or system failure characteristics may change to accomplish a different objective. The phases in a mission may be expressed in terms of: phase number, time interval, system configuration, task(s) to be undertaken, performance measure(s) of interest, and maintenance policy. This type of mission can be characterized as a sequence of discrete events required to complete a task e.g. an aircraft flight phase pattern.

In order to identify possible causes of phase and mission failure, a method is required to express how combinations of component failures (basic events) can occur during the phases throughout the mission and cause system failure. These failure events then require quantification to enable the likelihood and frequency of mission failure to be determined.

The main techniques that have previously been implemented for the solution to phased mission problems are that of Fault Tree Analysis, Markov Analysis and Simulation. The technique of *fault tree* analysis (FTA) is a commonly used tool to assess the probability of failure of industrial systems. This method may be adapted for analysis of systems comprising of more than one phase, where each phase depends on a different logic model. Hence the complexity of the modelling is significantly more difficult than for single phase systems. The fault tree approach represents the failure logic of the system in an inverted tree structure, and allows for both qualitative and quantitative system reliability analysis to take place. The earliest inspection of the analysis of phased missions was that carried out by Esary and Ziehms [**1**]. This research employed a fault tree method by which the mission is split into consecutive phases whereby each phase performs a specified task. The success of the mission depends on the performance of the non-repairable components used in each phase. The probability of this success is referred to as the *Mission Reliability*. Mission unreliability is defined as the probability that the system fails to function successfully during at least one phase of the mission. An important problem is to calculate, as efficiently as possible, either the exact value or bounds for the mission unreliability parameter. Methods to obtain estimates of such bounds are discussed by Burdick et al [**2**].

Situations may be encountered in phased mission analysis that prevent the assumption of independence between component failure or repair being made. In such circumstances, methods other than fault tree analysis must be applied. One such technique is the *Markov* approach [**3**]. The reliability of a mission may not be obtained by the simple multiplication of the individual phase reliabilities. This is due to the fact that at the phase change times, the system must occupy a state that allows both of the involved phases to function. The phases of the mission will be statistically dependent and an approach for solution has been presented by Smotherman and Zemoudeh[**4**] for repairable components. Of the many considered solutions to phased mission problems, simulation techniques typically offer the greatest generality in representation, but are also often the most expensive in computational requirements. The Markov method offers a combination of flexibility in representation and ease of solution but requires transition rates to be independent of time[**5**], and suffers from a potential explosion in the number of state equations for even moderate sized problems.

In some situations, it will be difficult to model a system by fault tree or Markov methods. This type of situation will occur if a system is too complex to use deterministic analysis, or if the failure and repair distributions of a component may not have a constant failure or repair rate. In such circumstances, simulation may be necessary.

Previous work has concentrated on assessing mission success. This paper identifies the probability of failure in each phase. Depending on the phase that the failure occurs, the consequences can be significantly different. Having calculated the probability of failure in each phase, the mission unreliability is simply the sum of the phase failure probabilities. In reducing the complexity of the problem in this way, the efficiency of the approach is improved. Further improvement can be achieved by employing modularisation methods and the Binary Decision Diagram (BDD) method. Focus will be restricted to a system where components are non-repairable.

2 Previous Fault Tree Methods for Phased Missions

A very simple phased mission problem consisting of non-repairable components with *A*, *B* and *C* representing component failures in each of the phases may be used to demonstrate approaches to phased misison analysis (Figure. 1).

The simple system will enable the features of the approaches to be understood without complicated analysis. During phase 1 which lasts until time t_1 the success of the mission is dependent upon all of the three components *A*, *B*, and *C*. Successful completion of phase 1 means the system then enters phase 2 which requires component *A* to function between times t_1 and t_2, along with at least one of the remaining two components *B* and *C*. The final phase requires that only one out of the three components must function between t_2 and t_3 for the mission to be accomplished successfully.

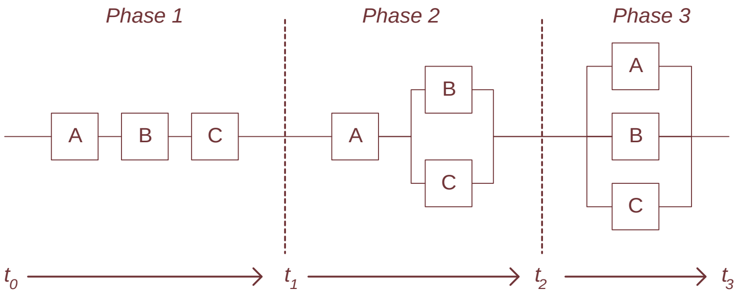


Figure 1 Reliability Network of a Simple Phased Mission System

Considering the phases as separate systems, the fault trees to represent individual phase failure are as shown in Figure 2. The notation used to represent component failure in phase *i* is A_i, B_i, C_i for components *A*, *B* and *C* respectively.

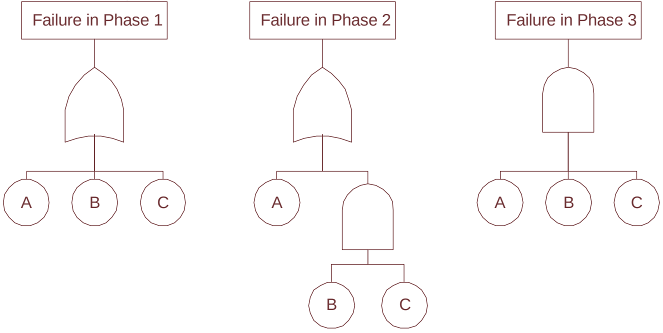


Figure 2 Fault Tree Representation of Individual Phase Failures

The minimal cut sets for each phase when treated as separate systems are:

Phase 1	Phase 2	Phase 3
A	A	$A B C$
B	$B C$	
C		

The method of calculating the reliability of a phased mission cannot simply be obtained by the multiplication of the reliabilities of each of the individual phases as this involves the false assumptions that the phases are independent and all components are in the working state at the beginning of each phase, and results in an appreciable over-prediction of system reliability.

A method proposed by Esary and Ziehms [**1**] involves the transformation of a multi-phase mission to that of an equivalent single phase mission. This transformation process involves three stages and is only concerned with the failure of the mission. It does not account for the phase in which failure occurs. Having expressed the failure causes for each phase by separate fault trees as in Figure 2, the transformation to single phased mission is achieved by:

1. Elimination of unnecessary cut sets. If cut sets of an earlier phase contain any from a later phase they may be removed from the first.

   For example if the minimal cut sets for each phase in the mission are:

Phase 1	Phase 2
AB	A
CDE	CF

   then minimal cut set AB can be removed from phase 1 as A failing in phase 1 means it will still be failed in phase 2 which will fail the mission. This makes the status of component B irrelevant.

   In the problem shown in figure 2 it means that minimal cut set A can be removed from phase 1.

2. Component failure events in each phase fault tree are replaced by an OR combination of the failure events for that and all preceding phases. For example, component A failure in phase 2 would be represented by the OR of the failure of the component in phase 1 (A_1) and in phase 2 (A_2) since the component is non-repairable (see Figure. 3). [Note the replacement is only performed on phase fault trees which have the eliminated minimal cut sets removed].

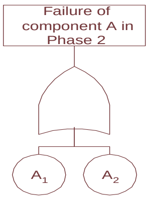


Figure. 3 Replacement OR combination

3. Each phase failure is combined using an OR gate to represent overall mission failure (i.e. the event that any phase does not complete successfully). This transforms the original multi-phase mission into an equivalent single phase mission as shown in Figure. 4.

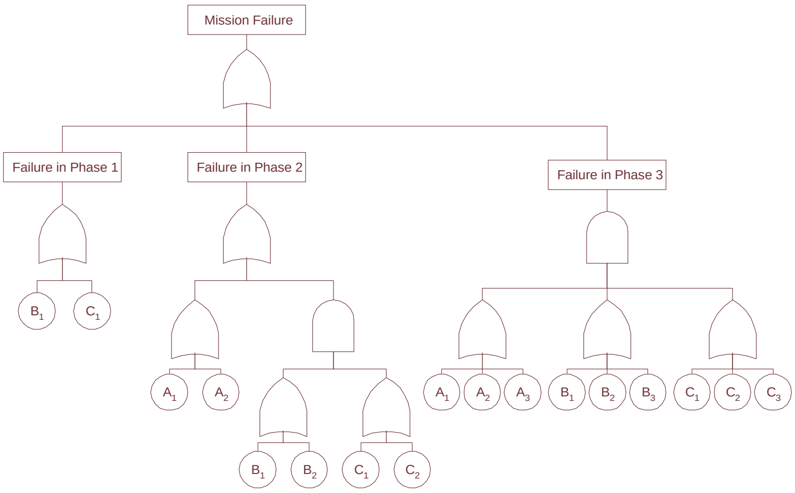


Figure. 4 Equivalent single phase mission

This equivalent single phase mission (see Figure. 4) produces different minimal cut sets than would have resulted from the combination of the individual phase minimal cut sets.

The process of removing cuts sets prior to the construction of fault trees can generally be seen to produce a simpler failure diagram and problem for analysis. However, since cuts sets are removed to produce this single phase mission, it becomes impossible to calculate individual phase failure probabilities which would be desirable.

3 Proposed Fault Tree Method for Phased Missions

A new method is proposed which enhances the fault tree approach in the previous section. It will enable the probability of failure in each phase to be determined in addition to the whole mission unreliability. For any phase, the method combines the causes of success of previous phases with the causes of failure for the phase being considered to allow both qualitative and quantitative analysis of both phase failure and mission failure.

The event of component failure in phase *i* is again represented as the event that the component could have failed during any phase up to and including phase *i*. System failure in phase *i* is represented by the AND of the success of phases *1..i-1* and the failure during phase *i*. (Figure. 5)

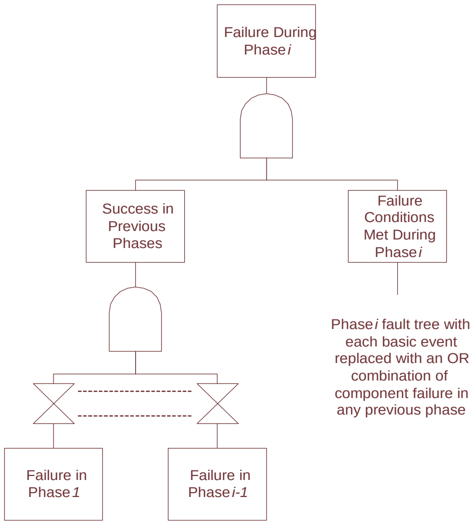

Failure During Phase *i*
Success in Previous Phases
Failure Conditions Met During Phase *i*
Phase *i* fault tree with each basic event replaced with an OR combination of component failure in any previous phase
Failure in Phase *1*
Failure in Phase *i-1*

Figure. 5 Generalised Phase Failure Fault Tree

Mission unreliability, Q_{MISS}, is then obtained from

$$Q_{MISS} = \sum_{i=1}^{n} Q_i \qquad (1)$$

where Q_i is the failure probability in phase *i* and *n* is the total number of phases.

This method allows for the evaluation of individual phase failures, and also accounts for the condition where components are known to have functioned to enable the system to function in previous phases. However, due to the fact that cut sets are not removed until a later stage in the analysis, the fault tree can be much more complex and require significantly more effort to solve.

The failure of a system may occur in many different ways. Each unique way is referred to as a *system failure mode*, and involves the failure of either a single component, or the combination of failures of multiple components. To determine the minimal cut sets of a phase or mission, either a top-down or a bottom-up approach is applied to the relevant fault tree. For any phases after the first phase, the incorporation of the success of previous phases means that the fault tree will be non-coherent and not simply consist of AND and OR gates. NOT logic will be required to represent this success, and the combinations of basic events that lead to the occurrence of the top event are referred to as *Prime Implicants*.

This proposed method may be applied for the simple three-phase mission given in Figure 1. The fault tree to represent the initial phase failure of the mission remains identical to the fault tree representation of the individual phase failure of phase *1* shown in Figure 2. Phase *2* failure can then be shown as the combination of phase *1* success and failure in phase *2* (Figure 6).

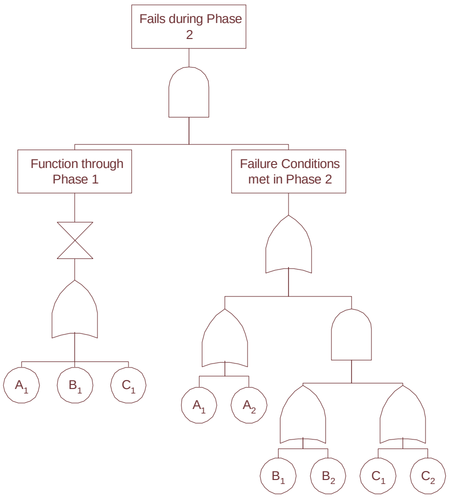


Figure 6 Phase 2 Failure Fault Tree

Similarly, phase *3* failure can be represented as the combination of phase *1* and phase *2* successes, and failure in phase *3* (Figure 7).

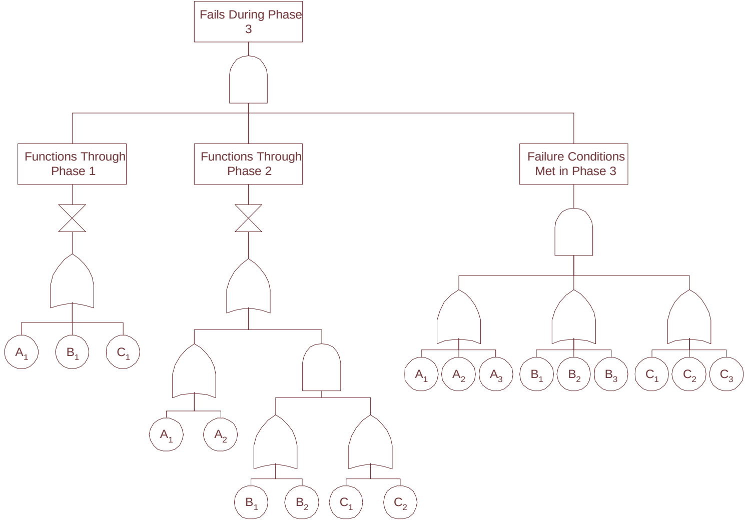


Figure 7 Phase 3 Failure Fault Tree

3.1 Fault Tree Modularisation

Fault tree modularisation techniques are helpful to reduce the size of a fault tree to enable prime implicants to be found more efficiently. These modularisation techniques reduce both memory and time requirements. A non-coherent extension of a modularisation technique developed by Riso [6] has been employed in this work. It repeatedly applies the stages of contraction, factorisation and extraction to reduce the complexity of the fault tree diagram. The phases are identified as:

- Contraction

Subsequent gates of the same type are contracted to form a single gate. The resulting tree structure is then an alternating sequence of OR and AND gates.

- Factorisation

Identification of basic events that always occur together in the same gate type. The combination of events and gate type is replaced by a complex event. However, since NOT logic is included in order to combine phase success and failure means that in this stage, the primary basic events that are found to always occur together in one gate type must have complements that always occur together in the opposite gate type by De Morgans's laws, e.g.

$$2000 = A + B \qquad 2001 = A \cdot B$$
$$\overline{2000} = \overline{A} \cdot \overline{B} \qquad \overline{2001} = \overline{A} + \overline{B}$$

- Extraction

Searches for structures within the tree of the form shown in Figure 8 that may be simplified by extracting an event to a higher level.

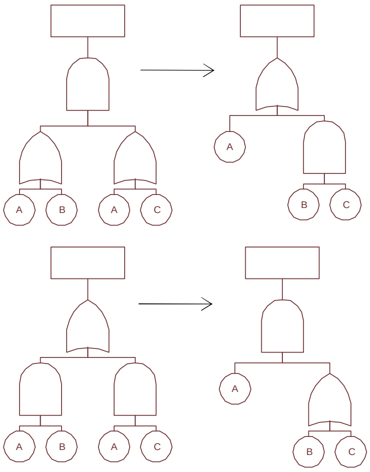


Figure 8 Extraction Stage of the Modularisation Technique

3.2 Prime Implicants in Phased Mission Systems

Due to the non-coherent nature of the fault trees, the combinations of basic events that lead to the occurrence of the top event of any phase failure are expressed as prime implicants. The notation used to represent the failure of component A in phase i is A_i . $\overline{A_i}$ represents the functioning of component A throughout phase i. The notation used to indicate the failure of a component in phase i to j is A_{ij} , i.e. fails at some time from the start of phase i to the end of phase j.

This notation enables us to define a new algebra over the phases to manipulate the logic equations. What is of concern in later phases is the time duration (i.e. phases) during which the component failures occur. So if we produce a combination of events for component A:

$$\overline{A_1}\;\overline{A_2}\;(A_3 + A_4)$$

it means we will only produce the top event being developed if A fails in phases 3 or 4 i.e. A_{34} where,

$$q_{A_{34}} = q_A(t_2, t_4) = \int_{t_2}^{t_4} f_A(t)dt$$

$f_A(t)$ is the density function of failure times for component A

A summary of the new algebraic laws are:

$$\begin{aligned} & A_i \cdot A_i = A_i \\ & A_i \cdot A_j = 0 \\ & A_i \cdot A_{ij} = A_i \\ & \overline{A_i} \cdot A_i = 0 \\ & \overline{A_i} \cdot A_{ij} = A_{i+1,j} \\ & \overline{A_i}\,\overline{A_{i+1}} \ldots \overline{A_j} = \overline{A_{ij}} \\ & A_i + A_{i+1} + \ldots + A_j = A_{ij} \end{aligned} \tag{2}$$

Therefore if two implicant sets contain exactly the same components where all but one of which occur over the same time intervals and the other is a failure in contiguous phases, the two implicant sets may be combined with the period of failure for the component with time discrepancy adjusted, eg:

$$\begin{array}{l} A_1 B_1 \\ A_1 B_2 \end{array} \xrightarrow{\qquad\qquad} A_1 B_{12}$$

As the components are non-repairable, the event of component failure will only be possible over contiguous phases. This simplification approach therefore allows the prime implicants for the simple example given in Figure 1 to be expressed as:

<u>Phase 1</u>

$$T_1 = A_1 + B_1 + C_1$$

Minimal Cut Sets: A_1, B_1, C_1

<u>Phase 2</u>

$$\begin{aligned} T_2 &= \overline{A_1}\,\overline{B_1}\,\overline{C_1}(A_1 + A_2 + B_1C_1 + B_1C_2 + B_2C_1 + B_2C_2) \\ &= \overline{A_1}A_2\overline{B_1}\,\overline{C_1} + \overline{A_1}\,\overline{B_1}\,\overline{C_1}B_2C_2 \\ &= A_2\overline{B_1}\,\overline{C_1} + \overline{A_1}B_2C_2 \end{aligned}$$

Prime Implicants: $A_2\overline{B_1}\,\overline{C_1}$, $\overline{A_1}B_2C_2$

<u>Phase 3</u>

$$\begin{aligned} T_3 &= \overline{A_1}\,\overline{B_1}\,\overline{C_1}((\overline{A_1}\,\overline{A_2}(\overline{B_1}\,\overline{B_2} + \overline{C_1}\,\overline{C_2}) \cdot (A_1 + A_2 + A_3) \cdot (B_1 + B_2 + B_3) \cdot (C_1 + C_2 + C_3) \\ &= A_3B_3C_{23} + A_3B_{23}C_3 \end{aligned}$$

Prime Implicants: $A_3B_3C_{23}$, $A_3B_{23}C_3$

3.3 Quantification

Having established the prime implicants for each phase they may now be used to quantify the probability of phase and mission failure.

The unreliability, Q_i ,for each individual phase *i* is found using a simple inclusion-exclusion expansion for the prime implicants C_j in the phase,

$$Q_i = \sum_{j=1}^{N_i} P(C_j) - \sum_{j=1}^{N_i} \sum_{k=1}^{j-1_i} P(C_j \cap C_k) + \cdots + (-1)^{N_i - 1} P(C_1 \cap C_2 \cap \cdots \cap C_{N_i})$$

Therefore the event of phase failure for this simple three-phase mission may be expressed as:

Phase 1: $Q_1 = q_{A_1} + q_{B_1} + q_{C_1} - q_{A_1} q_{B_1} - q_{A_1} q_{C_1} - q_{B_1} q_{C_1} + q_{A_1} q_{B_1} q_{C_1}$

Phase 2: $Q_2 = q_{A_2}(1 - q_{B_1})(1 - q_{C_1}) + (1 - q_{A_1}) q_{B_2} q_{C_2} - q_{A_2} q_{B_2} q_{C_2}$ (3)

Phase 3: $Q_3 = q_{A_3} q_{B_3} q_{C_{23}} + q_{A_3} q_{B_{23}} q_{C_3} - q_{A_3} q_{B_3} q_{C_3}$

As the failure of each of the phases produce mutually exclusive causes, the probability of mission failure may be expressed as a sum of the unreliabilities of the individual phases:

$$Q_{MISS} = \sum_{i=1}^{n} Q_i \qquad (4)$$

For systems with non-repairable components, the expected number of failures per mission is equal to the mission unreliability.

4 Binary Decision Diagram Analysis for Phased Missions

A Fault Tree Structure is very efficient to represent system failure logic, however is not an ideal form for mathematical analysis. Binary Decision Diagrams (BDDs) represent a logic expression and offer efficient mathematical manipulation, although it is very difficult to construct directly from the system definition. For larger fault trees it is more efficient to convert to a BDD prior to analysis. The approach of performing the quantification process after first converting the fault tree to a BDD form offers significant advantages for large complex fault trees. This is particularly true of structures which are non-coherent, such as the phase failure fault trees.

Figure 9 shows a Binary Decision Diagram. Paths through the BDD start at the top root node and terminate at one of two terminal nodes, 1 or 0. A terminal one indicates top event occurrence and terminal zero is top event non-occurrence.

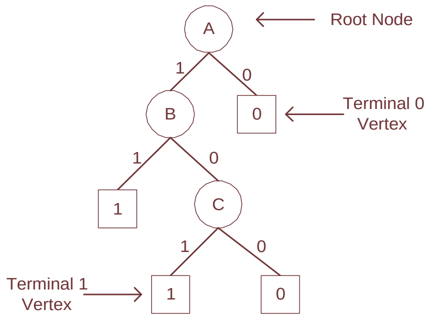


Figure 9 Binary Decision Diagram

Each node on the diagram corresponds to a basic event in the fault tree which has to be placed in an ordering prior to BDD construction. In this case the ordering is $A<B<C$.

All nodes have two out branches, a *1* branch corresponds to component failure and a *0* branch corresponds to a component working state.

Prime implicants are given by events on paths through the diagram which lead to a terminal *1* vertex, i.e.,

$$AB$$

$$A\bar{B}C$$

In this case, by consensus these can be reduced to minimal cut sets:

$$AB$$

$$AC$$

since this BDD does not represent a non-coherent system. More details of qualitative BDD analysis can be found in reference [**7**].

4.1 Construction

A binary decision diagram consists of vertexes where each vertex has an if-then-else structure as shown in Figure 10.

```
If(X1) then
        Consider function f1
else
        Consider function f2
endif
```

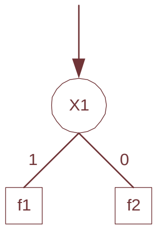


Figure 10 A Binary Decision Diagram Vertex

This if-then-else structure is represented in shorthand (*ite*) notation as

$$ite(X1, f1, f2)$$

To combine two basic events using a logical operation $\oplus$,

$$\begin{array}{ll} \text{If} & J = ite(X, f1, f2) \\ \text{and} & H = ite(Y, g1, g2) \end{array}$$

$$\text{If } X < Y \qquad J \oplus H = ite(X, f1 \oplus H, f2 \oplus H)$$
$$\text{If } X = Y \qquad J \oplus H = ite(X, f1 \oplus g1, f2 \oplus g2) \tag{5}$$

The basic events in any BDD are represented as $A = ite(A,1,0)$ and $\overline{A} = ite(A,0,1)$. For phased mission systems, the BDD is constructed so that each component is considered in each phase (in phase order) before the next basic event is taken into account.

4.2 Quantification

Since each path to a terminal *1* is disjoint then the exact top event probability Q_{EXACT} is given by:

$$Q_{EXACT} = \sum_{i=1}^{n} p(r_i)$$

where $p(r_i)$ is the probability of the *i*th disjoint path to $\boxed{1}$

Further details of BDD quantification can be found in reference [**8**].

4.3 Example

The simple three-phase mission illustrated in Figure 1 may be represented in BDD form for each phase and then quantified. The fault trees for phase 1 (Figure 2), phase 2 (Figure 6) and phase 3 (Figure 7) are first converted to BDDs. These BDDs are shown in Figures 11-13 respectively. Their analysis is presented below.

In BDD methodology to evaluate the success of a phase as opposed to the failure, a *1* is replaced by a *0* and a *0* by a *1* for the terminal nodes.

Phase 1

For phase 1, the *ite* structure represented by the BDD in Figure 11 is:

$$ite(A_1,1,ite(B_1,1,ite(C_1,1,0)))$$

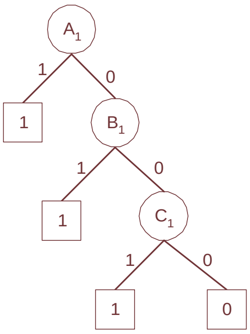


Figure 11 Phase 1 Failure BDD

Analysis of this BDD gives

Minimal Cut Sets: A_1
B_1
C_1

$$Q_1 = q_{A_1} + (1 - q_{A_1})q_{B_1} + (1 - q_{A_1})(1 - q_{B_1})q_{C_1}$$

Phase 2

The BDD for failure during phase 2 is given by the following *ite* structure:

$$ite(A_1,0,ite(A_2,ite(B_1,0,ite(C_1,0,1)),ite(B_1,0,ite(B_2,ite(C_1,0,ite(C_2,1,0)),0))))$$

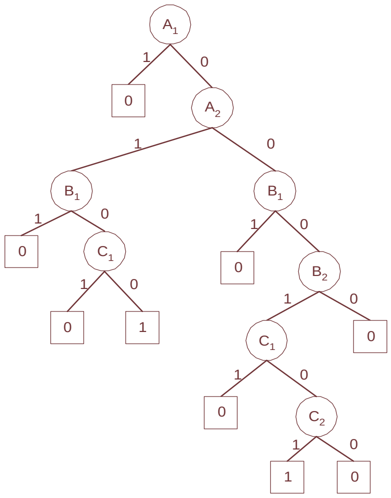


Figure 12 Failure During Phase 2 BDD

For each path to a terminal *1*, using the algebra of events gives,

$$\text{Prime Implicants:} \quad \begin{array}{lcl} \overline{A_1}\ A_2\ \overline{B_1}\ \overline{C_1} & \Rightarrow & A_2\ \overline{B_1}\ \overline{C_1} \\ \overline{A_1}\ \overline{A_2}\ \overline{B_1}\ B_2\ \overline{C_1}\ C_2 & & \overline{A_{12}}\ B_2\ C_2 \end{array}$$

Nodes on a BDD path will represent failure or functioning of a particular component through different phases. These must be combined using the algebra of events given earlier prior to evaluating the probability of the status required of that component.

Having considered each component encountered on a path the probability of the path to the terminal *1* is evaluated as usual by taking the product of the probability of the component status. The phase failure is then obtained by summing the probability of each disjoint path.

$$Q_2 = q_{A_2}(1-q_{B_1})(1-q_{C_1}) + (1-q_{A_1}-q_{A_2})q_{B_2}q_{C_2}$$

Phase 3

The BDD representation for the fault tree representing failure during phase 3 is:

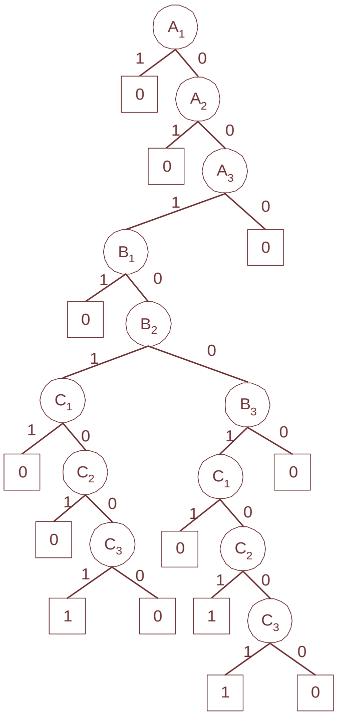


Figure 13 Failure during Phase 3 BDD

$$\text{Prime Implicants:}\quad \begin{matrix} \overline{A_1}\ \overline{A_2}\ A_3\ \overline{B_1}\ \overline{B_2}\ B_3\ \overline{C_1}\ \overline{C_2}\ C_3 \\ \overline{A_1}\ \overline{A_2}\ A_3\ \overline{B_1}\ \overline{B_2}\ B_3\ \overline{C_1}\ C_2 \\ \overline{A_1}\ \overline{A_2}\ A_3\ \overline{B_1}\ B_2\ \overline{C_1}\ \overline{C_2}\ C_3 \end{matrix} \longrightarrow \left.\begin{matrix} A_3B_3C_3 \\ A_3B_3C_2 \\ A_3B_2C_3 \end{matrix}\right\} \longrightarrow \begin{matrix} A_3B_3C_{23} \\ A_3B_{23}C_3 \end{matrix}$$

$$Q_3 = q_{A_3}q_{B_3}q_{C_{23}} + q_{A_3}q_{B_{23}}q_{C_3} - q_{A_3}q_{B_3}q_{C_3}\ .$$

Therefore it can be seen that the unreliability of each of the phases as found by the BDD method is identical to that obtained using fault tree analysis (Equation 3).

5 Conclusions

1. The accurate assessment of mission unreliability for systems with non-repairable components operating over a sequence of phases can be performed using non-coherent fault tree structures.

2. The direct quantification of the fault trees is frequently problematic for even moderate sized problems due to the size and complexity of the resulting logic functions.

3. Fault tree modularisation methods provide some reduction in the size of the problem but not enough for this alone to offer a practical solution method.

4. The use of Binary Decision Diagrams (enhanced to account for the phased nature of component failures) to calculate the failure probability of each phase in the mission provides an efficient and accurate means of evaluating the mission reliability.

6 References

[1] J.D.Esary, H. Ziehms, "Reliability of phased Missions", *Reliability and Fault-Tree Analysis*, Society for Industrial Applied Mathematics, Phila. 1975, pp213-236, 1974 September.

[2] G.R.Burdick, J.B.Fussell, D.M.Rasmuson, J.R.Wilson, "Phased Mission Analysis: A Review of New Developments and an Application", *IEEE Trans. Reliability*, vol R-26, 1977 Apr, pp 43-49.

[3] C.A.Clarotti, S.Contini, R.Somma, "Repairable Multiphase Systems - Markov and Fault-Tree Approaches for Reliability Evaluation", in Apostolakis, Garribba, Volta (eds.), *Synthesis and Analysis Methods for Safety and Reliability Studies*, Plenum Press, 1980, pp 45-58.

[4] Mark Smotherman, Kay Zemoudeh, "A Non-Homogeneous Markov Model for Phased Mission Reliability Analysis", *IEEE Trans. Reliability*, vol 38, 1989 Dec, pp 585-590.

[5] Mark Smotherman, Robert M. Geist, "Phased Effectiveness using a Non-homogeneous Markov Reward Model", *Reliability Engineering and System Safety*, vol 27, 1990, pp 241-255.

[6] K. Reay, J.D. Andrews, "A fault tree analysis strategy using binary decision diagrams", *Reliability Engineering and System Safety*, vol 78, 2002, pp45-56.

[7] R.M. Sinnamon, J.D. Andrews, "Improved efficiency in qualitative fault tree analysis", *Quality and Reliability Eng'g Int'l*, vol 13, num 5, 1997, pp293-298.

[8] R.M. Sinnamon, J.D. Andrews, "Improved accuracy in qualitative fault tree analysis", *Quality and Reliability Eng'g Int'l*, vol 13, num 5, 1997, pp285-292.

Acknowledgement

The work described in this paper was conducted as part of a research project funded by the Ministry of Defence. The views expressed are those of the authors and should not be considered as those of the Ministry of Defence.
The authors would like to thank Richard Denning, MOD, for his input to the research described in the paper.