NASA Contractor Report 191564
ICASE Report No. 93-84

ICASE

PHASED-MISSION SYSTEM ANALYSIS USING BOOLEAN ALGEBRAIC METHODS

Arun K. Somani
Kishor S. Trivedi

NASA Contract No. NAS1-19480
November 1993

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23681-0001

Operated by the Universities Space Research Association

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23681-0001

PHASED-MISSION SYSTEM ANALYSIS USING BOOLEAN ALGEBRAIC METHODS

Arun K. Somani[1]
Dept. of Electrical Eng. and Dept. of Computer Sci. and Eng.
University of Washington, FT-10
Seattle, WA 98195

and

Kishor S. Trivedi[1]
Department of Computer Science and Engineering
Duke University
Durham, NC, 27708

ABSTRACT

Most reliability analysis techniques and tools assume that a system is used for a mission consisting of a single phase. However, multiple phases are natural in many missions. The failure rates of components, system configuration, and success criteria may vary from phase to phase. In addition, the duration of a phase may be deterministic or random. Recently, several researchers have addressed the problem of reliability analysis of such systems using a variety of methods. We describe a new technique for phased-mission system reliability analysis based on Boolean algebraic methods. Our technique is computationally efficient and is applicable to a large class of systems for which the failure criterion in each phase can be expressed as a fault tree (or an equivalent representation). Our technique avoids state space explosion that commonly plague Markov chain-based analysis. We develop a phase algebra to account for the effects of variable configurations and success criteria from phase to phase. Our technique yields exact (as opposed to approximate) results. We demonstrate the use our technique by means of an example and present numerical results to show the effects of mission phases on the system reliability.

[1]This research was supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-19480 while the authors were in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23681.

1 Introduction

The reliability analysis of ultra-reliable computer systems is an important problem for which various techniques and tools have been developed [1]-[4]. Most analysis techniques assume that the systems operate in single-phase missions. However, multiple phases are natural in many applications. The system configuration, operational requirements for individual components, the success criteria, and the stress on the components (and thus the failure rates) may vary from phase to phase. For example, fault tolerant systems may consist of multiple subsystems employing redundancy and may have dedicated or pooled spares. A dedicated spare can replace only a single preassigned function. A pooled spare, on the other hand, has the capability of replacing any of the several functions in the system. Depending on the requirements during different phases, spares may be placed in service or removed from service to balance the system reliability and the cost of operation. The success of a redundancy management scheme defines if a system is operational or not. The usage of subsystems may also vary from phase to phase and subsystems supporting those services may remain idle or may be switched off. Furthermore, the duration of any phase may be deterministic or random. All these variations affect the system reliability.

Sometimes the effects of phased missions can be ignored in favor of simpler analysis. For example, in an airplane system, landing gear and its associated control subsystems are not required during cruising phase. So exact analysis should not ignore such failures. But, continuing to count the failure of landing gear during cruising phase has very little impact on the overall unreliability and may simplify the computation. However, most of the time only conservative estimates can be made, thus yielding the worst case unreliability of the system. One adverse effect of this is that the systems are over-designed. For economic reasons, it may be desirable to perform more accurate analysis. In particular, if one phase may see much more stress than others then it is necessary to account for these effects properly. It is not accurate to use conservative parameters for the the entire mission. On the other hand the impact of a phase with severest parameter values must not be ignored in analysis. Different aspects of phased-mission systems have been discussed by several researchers.

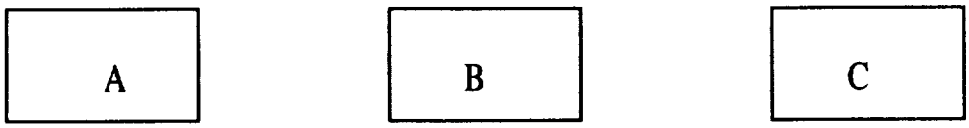


Figure 1: The three units in a system

To describe and compare the work here of others and our own, we will use a three component system as an example. Components A, B, and C are used in a system which is employed in a mission with 3 phases. The phases are denoted as Phase X, Phase Y, and Phase Z, respectively. To show the effect of phased-mission analysis we will consider all six permutation of these three phases. That is, we will assume that the mission may go through the three phases in any order. So one particular order may be Phases X, Y, and Z or another

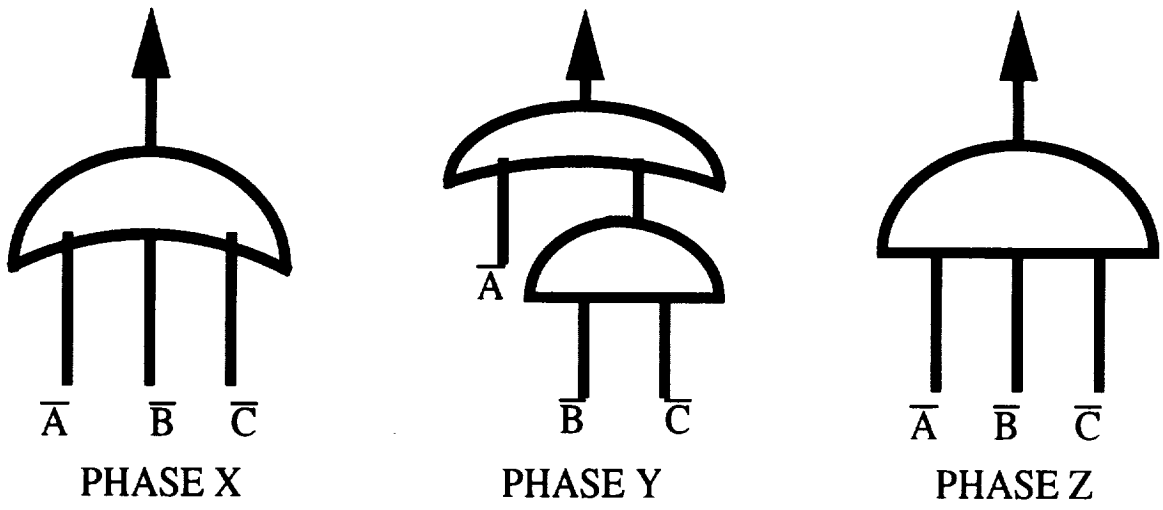


Figure 2: The success criteria for phases expressed using fault trees

could be Phases Z, Y, and X. The success criteria for each of the three phases is expressed using fault trees as shown in Figure 2. In Phase X, the system fails if any of the components A, B, or C fails. In Phase Y, the system fails if component A fails or both of the components B and C fail. In Phase Z, the system fails if all three components fail. The failure rates of three components are λ_a, λ_b, and λ_c, respectively.

The corresponding Markov chains for all phases are shown in Figure 3. In the Markov chain representation, a 3-tuple represents a state indicating the status of the three components respectively. A "1" represents that the corresponding component is alive and a "0" represent that the component has failed. For example, a state (101) implies that component B has failed and the other two components are alive. A transition from one state to another state has a rate associated with it which is the failure rate of the component that fails. For example, a transition from state (011) to state (010) has a transition rate of λ_c. States marked F are failed states.

2 Related Work

Esary and Ziehms [5] discuss analysis of multiple configuration systems during different phases of a mission to accomplish specified goals. In their approach, each phase of a system is modeled using a separate reliability block diagram (RBD). For phase p, a component C is represented by a series of a blocks $c_1, c_2, \cdots, c_p$ where c_i represents the probability of failure (or success) associated with component C in a phase i and depends on the failure rate of that component during that phase. All phase RBDs are connected in series as shown in Figure 4 for a three phase system using three components. Solution of this RBD correctly predicts the reliability of the three phase system. The problem with this approach is a large RBD with several common events, the solution of which may be computationally very expensive. Each component generates p basic event for a p-phased system. A k component system will thus have $k * p$ basic events and obtaining cut

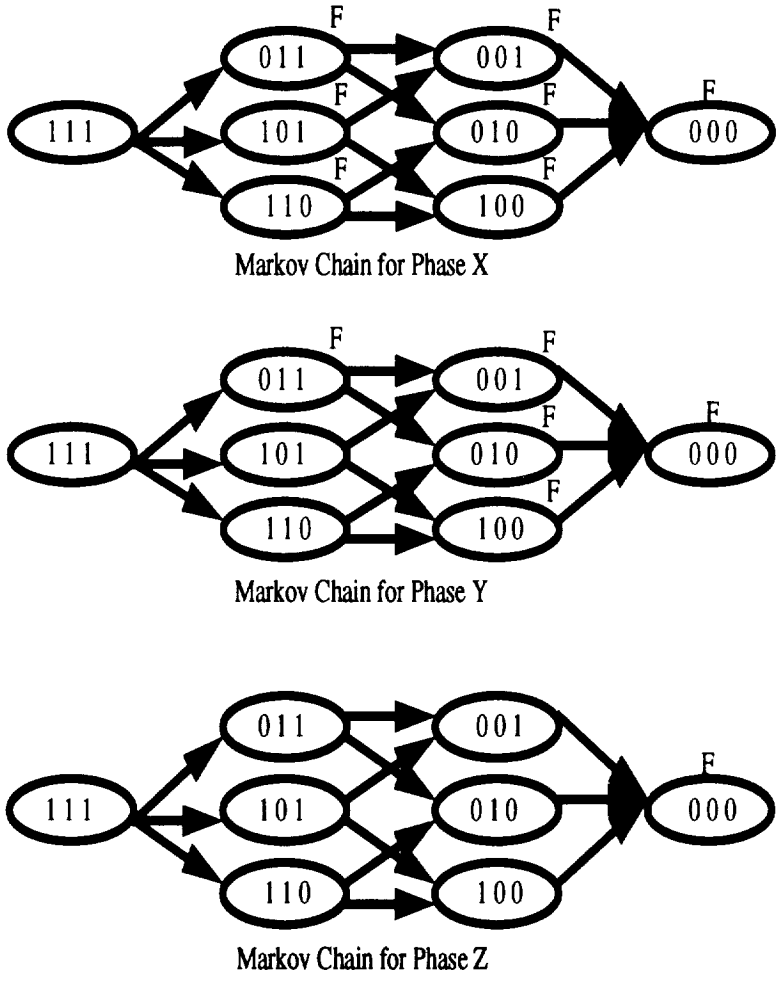


Figure 3: The Markov chains for three phases

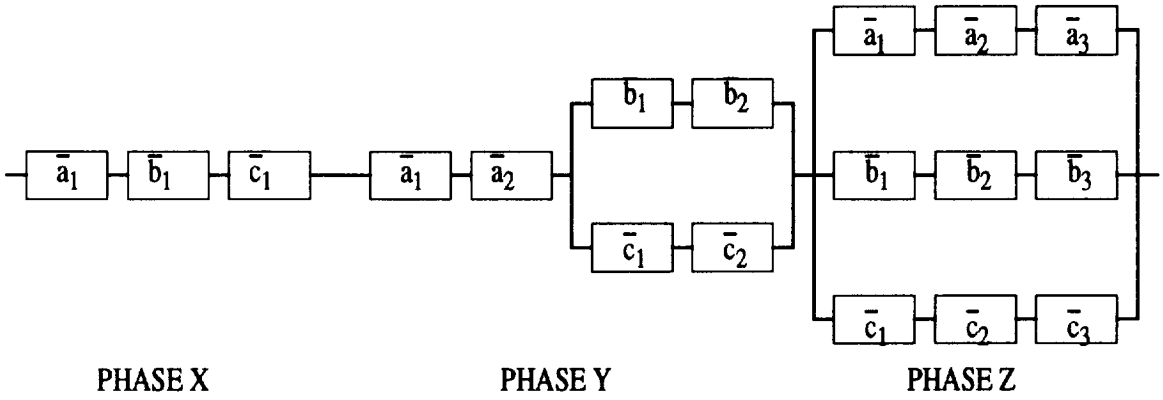


Figure 4: Reliability block diagram for a three phases system with variable configuration

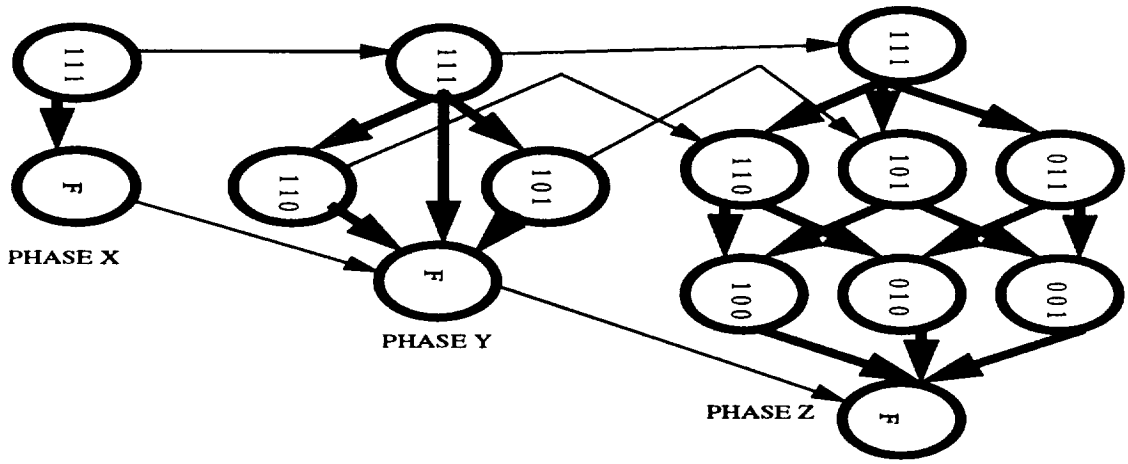


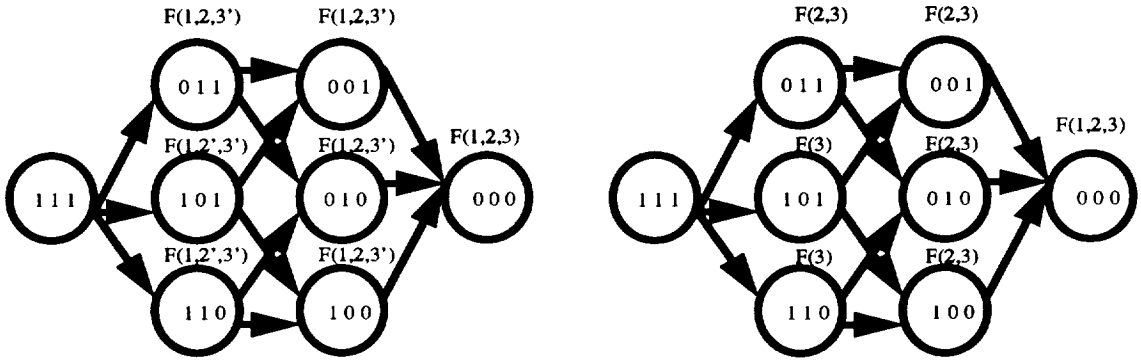
Figure 5: The multi-phase Markov chain

sets after accounting for common events will be expensive. Approximate solution to RBD may include large errors due to multiple common events.

Pedar and Sarma [6] carry out phased-mission analysis of an aerospace computing systems using an approach similar to Esary and Ziehms. They developed a procedure to systematically cancel out the common events in earlier phases which are accounted for in later phases. Alam and Al-Saggaf [7] developed a technique to analyze repairable systems in which system success criteria and failure rates of components may vary from phase to phase.

Smotherman and Zemoudeh [9] use a non-homogeneous Markov model to carry out a phased-mission system analysis. They represent the behavior of the system in each phase using a different Markov chain and each phase is represented by a separate subset of the states. The state transitions, which are described in terms of random variables, are generalized to include phase changes. Therefore, state dependent phase changes, random phase durations, time varying failure and repair behavior are readily modeled. A complete Markov chain of a three phase system of Figure 2 with phase order of X, Y, and Z is shown in Figure 5. The major drawback of this approach, like Esary and Ziehms approach using RBDs, is a huge non-homogeneous Markov chain. The size of the state space is as big as the sum of the number of states in each of the individual phase. This requires large amount of storage and computation time to solve a system limiting the kind of systems that can be analyzed.

Somani et. al. [10] presented a computationally efficient method to analyze multi-phased systems and a new software tool for reliability analyses of such systems. A system with variable configuration and success criteria results in different Markov chains for different phases as shown in Figure 5. In Somani et. al.'s approach, instead of a single Markov chain, Markov chains for individual phases are developed and solved separately. The issue of varying success criteria and change in system configuration from phase to phase



(a) Markov Chain for Phase X, Y, and Z

(b) Markov Chain for Phases Z, Y, and X

Figure 6: Two scenarios for phased-mission systems with variable configuration

is addressed by providing an efficient mapping procedure at the transition time from a phase to another phase. While analyzing a phase, only the states relevant to that phase, are considered. Thus each individual Markov chain is much smaller than in Smotherman and Zemoudeh [9]. For example, in Figure 5, three Markov chains with number of states 2, 4, and 8, respectively are solved instead of a single Markov chain with 12 states. Using this approach, the computation time for large systems can be reduced significantly without compromising accuracy. Phases may be of a fixed or a random duration. The reliability (or unreliability) of the system can be computed from the output of final phase. Furthermore, the technique is sufficiently general.

Using a similar approach, Dugan [8] suggested another method in which a single Markov chain with state space equal to the union of the state spaces of the individual phases is generated. The transitions rates are parameterized with phase numbers and the Markov chain is solved p times for p phases. The final state occupation probabilities of one phase become the initial state occupation probabilities for the next phase. In her approach, once a state is declared a system down state in a phase, it cannot become an up state in a later phase. This is a potential problem as it is possible for a system to have some states that are failure states in a phase but are up states in a later phase. For example, consider the two scenarios as shown in Figure 6. In the first case (Figure 6a), phase order is Phase X, Phase Y, and Phase Z. In this case, some of the states are failure states in the first phase that are later on treated as forced failure states although they are not failure states in phases 2 and 3. Such states are marked as F(1,2',3') or F(1,2,3'). In the second case, phase order is Phase Z, Phase Y, and Phase X. In this case, there are no forced failure states.

In this paper, we present a methodology to analyze and solve phased-mission systems in which failure rates, configuration and success criteria can vary from phase to phase. Moreover, the success criteria can be specified using fault trees or an equivalent representation. We believe that a majority of systems can be represented using fault trees. Our approach is similar to Esary and Ziehms' in that we do not generate any Markov chains, but in addition we do not create a single, monolithic model. We handle one phase at a time and then compute the overall unreliability of the entire mission. This gives us a computational advantage.

First we describe some concepts which we will use throughout the paper.

3 Distribution Functions with Mass at Origin

One of the key concepts we will use in our method is that of cumulative distribution functions with a mass at the origin. Consider a random variable X with cumulative distribution function given by

$$F_X(t) = (1 - e^{-\lambda T_1}) + e^{-\lambda T_1}(1 - e^{-\lambda t}).$$

This function has a mass at the origin given by $P(X = 0) = (1 - e^{-\lambda T_1})$. The second term represents the continuous part of the distribution function.

In order to illustrate the use of such a CDF, consider a component with a failure rate of λ that is used in a phased mission system. Assume that the system has just completed one phase of duration T_1 and is currently in the second phase. The above CDF can be assigned as the failure probability distribution of the component in the second phase. The first term in the above expression represents the probability that the component has already failed in the previous phase. The second term represents the failure probability distribution for this component for the second phase. The time origin for the second phase is reinitialized to the beginning of the phase. We will use such distribution functions to represent failure probabilities of individual components during different phases.

4 Phased-Mission Analysis: Phase Independent Success Criteria

In this section we consider a simpler scenario, a phased-mission system in which the success criterion is phase independent. Therefore, the system configuration and the success criteria remains unchanged from phase to phase and can be represented by the same fault tree for all phases. However, component failure rates are allowed to be phase dependent. We first assume that phase durations are deterministic. We will relax these constraints one at a time in the following subsections.

4.1 Phase-Dependent Failure Rates

To account for phase-dependent failure rates, we assign a failure distribution with mass at the origin to each component. Let λ_{ji} represent the failure rate of component j in phase i. For component j, the distribution function assigned in phase k is given by

$$F_{C_{j,k}}(t) = (1 - e^{-\sum_{i=1}^{k-1} \lambda_{ji} T_i}) + e^{-\sum_{i=1}^{k-1} \lambda_{ji} T_i}(1 - e^{-\lambda_{jk} t}). \quad (1)$$

Here time t is measured from the beginning of phase k so that $0 \leq t \leq T_k$. T_i represents the duration for phase i. This expression can be simplified to: $F_{C_{j,k}}(t) = 1 - e^{-\lambda_{jk} t}[e^{-\sum_{i=1}^{k-1} \lambda_{ji} T_i}]$. At the end of phase k,

at $t = T_m$, the above expression gives the mass at the origin for phase $k+1$. A component fails during a phase only if it survives during all the previous phases. The factor enclosed in square brackets above is the probability of success during first $k-1$ phases. Since the success criteria is same in all phases, a system fails by phase k if it fails any time during the first k phases. We can obtain the unreliability of the system at time $0 \leq t \leq T_k$ during phase $1 \leq k \leq m$ by evaluating the fault tree using the failure distribution function for each component as given by $F_{C_{j,k}}(t)$. Of course, if our only interest is in the failure probability for the entire mission, we evaluate the fault tree assigning a constant faliure probability

$$1 - e^{-\sum_{i=1}^{i=m} \lambda_{ji} T_i}.$$

to component j .

4.2 Age-Dependent Failure Rates

If the failure rates of components are phase and age dependent then we cannot count time for each phase independently. Instead, to compute the failure probability distribution, we have to account for the global (mission) time and its affect on each component. This can be achieved by assigning the failure distribution function for component j in phase k as follows.

$$F_{C_{j,k}}(t) = (1 - e^{-\sum_{i=1}^{k-1} \int_{CT_{i-1}}^{CT_i} \lambda_{ji}(\tau)d\tau}) + e^{-\sum_{i=1}^{k-1} \int_{CT_{i-1}}^{CT_i} \lambda_{ji}(\tau)d\tau} (1 - e^{-\int_{CT_{k-1}}^{t} \lambda_{jk}(\tau)d\tau}).$$

Here,

$$CT_i = \sum_{l=1}^{i} T_l$$

is the sum of durations for i phases and $CT_0 = 0$. The time t is the cumulative time and is not reset to zero for the next phase. Instead it starts at $t = 0$ at the beginning of a mission and continues to increase. With this modification, the fault tree can be evaluated for any time $0 \leq t \leq CT_m$. The probability of failure of component C_j at the end of the mission is given by

$$1 - e^{-\sum_{i=1}^{m} \int_{CT_{i-1}}^{CT_i} \lambda_{ji}(t)dt}$$

Using this constant failure probability for component C_j (for all j), the fault tree can be evaluated to obtain the mission failure probability.

4.3 Random Phase Durations

To account for random phase durations, we use conditioning followed by the theorem of total probability. Let $F_{T_i}(t_i)$ be the distribution function for the length of phase i. These distributions are specified by the

user. Conditioning on the durations of phases $T_1 = t_1, T_2 = t_2, \cdots, T_m = t_m$ the mission failure probability for component j is given by

$$1 - e^{-\sum_{i=1}^{i=m} \lambda_{ji} t_i}.$$

Then the unconditional faliure probability for component j is given by

$$\int\int \ldots\ldots \int [1 - e^{-\sum_{i=1}^{i=m} \lambda_{ji} t_i}] dF_{T_1}(t_1) \ldots\ldots dF_{T_m}(t_m) = 1 - \Pi_{i=1}^{m} F_{T_i}^{\sim}(\lambda_{ji})$$

where $F_{T_i}^{\sim}(s)$ is the LST (Laplace Stieltejs transform) of T_i so that $F_{T_i}^{\sim}(s) = \int_0^{\infty} e^{-st_i} dF_{T_i}(t_i)$

This failure probability can be assigned to component C_j (for all j) and the fault tree can be evaluated to compute the unreliability of the system for the whole mission consisting of m phases.

5 Phased-Mission Analysis: Phase-Dependent Success Criteria

The results of the previous section apply to the cases when the success criteria does not change from phase to phase. However, in many applications, the success criteria and the system configuration may change from phase to phase. There are several reasons for reconfiguration and change in success criteria from phase to phase. Some of these are discussed below.

1. A component is used in all phases but its operational level requirements may change. In this case, no special treatment is required for this component. The definition of operation or failed state depends on the success criteria.

2. A component is used in a n consecutive phases starting with some phase k, and is then not needed for system operation in the remaining phases.

3. A component is required to remain operational for some phase, is not need for the operation of a few phases and is then required again for system operation.

4. Additional redundant modules are added during the operation of the system.

5. Some redundant modules are removed from a subsystem.

6. Spare or operational redundant modules corresponding to one subsystem become spare or redundant modules for another subsystem.

Due to a change in success criterion, it is possible that some combination of failures of components in one phase leads to failure of the system whereas the same combination does not lead to failure in some other phase. In Markov chain-based methods, it is easier to keep track of the system states, and therefore, change in system success criteria could be easily accounted for. However, in the case of a fault tree, this change

needs to be accounted for by considering cases when the system may fail or may not fail at the time of phase transition. There are four possible cases which may occur at the time of a phase transition from phase i to phase $i+1$.

1. A combination of component failures does not lead to system failure in both phases i and $i+1$.
2. A combination of component failures leads to system failure in both phases i and $i+1$.
3. A combination of component failures does not imply system failure in phase i but is treated as system failure in phase $i+1$.
4. A combination of component failures implies system failure in phase i but does not imply system failure in phase $i+1$.

The first two cases require treatment similar to that in the previous section as the success criteria does not change from phase i to phase $i+1$ with respect to the failure combination under consideration. Failure combinations in the third case above should be treated as failures in the earlier phase i as well. This is because such combinations, once present during a phase are bound to lead to the system failure eventually at the transition time when the systems enters this later phase. These are referred to as latent failures in [11]. Hence a more stringent criterion should be applied with respect to these combinations. So we can assume that all failure combinations in phase $i+1$ are also failure combinations in phase i (but not vice versa). Hence for the first three cases, the unreliability can be evaluated by evaluating the fault tree for the last phase using the approach of Section 4.

The failure combinations which imply system failure in phase i, but do not lead to system failure in subsequent phases, as is the fourth case, should be handled more carefully. We need to account for the probability of occurrence of these failure combinations until phase i. Any probability attributed to such combinations of component failures in later phases does not contribute towards system unreliability. Esary and Ziehms account for this by cascading the phase reliability blocks. However, as mentioned earlier, that leads to a more expensive computation. We present our method of handling such failure combinations below.

Our methodology consists of the following steps. We divide the system unreliability of a phased mission system into two parts: (i) common failure combinations; and (ii) phase failure combinations. We evaluate the unreliability due to these two components using the following procedure.

5.1 Common Failure Combinations

The first component, common failure combinations, includes the probability of those component failure combinations which are common to all phases after the most stringent criterion has been applied to all phases. That is, if a combination leads to system failure in phase $i+1$, then it is a considered a failure combination in

phase i as well. Thus the common failure combinations essentially include the failure combination specified for the last phase.

The unreliability due to common failure combinations can be computed using the method described in the previous section for analyzing phased-mission system with phase-independent success criteria. That is, we compute the failure probability distribution for individual component and then evaluate the common fault tree which is the fault tree for the last phase.

5.2 Phase Failure Combinations

The second component, phase failure combination, includes the probability of all failures specific to individual phases after applying the most stringent success criterion in each phase. For phase i, this part include the probability of only those component failure combinations which contribute to system failure in phase i but are considered operational in all subsequent phases.

Unreliability due to the second component requires additional computations. For each phase, we need to identify and compute the probability of component failure combinations which lead to system failure in that phase and does not imply system failure in any subsequent phase. Let E_i be the Boolean logic expression specifying the failure combinations for phase i. Then phase failure combinations for phase i (PFC_i), which are treated as success combinations for the all subsequent phases are given by

$$PFC_i = (\cdots((E_i \wedge \overline{E_{i+1}}) \wedge \overline{E_{i+2}}) \cdots \wedge \overline{E_p}).$$

In the above expression, we include only those combinations which are failure combinations in phase i but are not failure combinations in any of the subsequent phases. This expression can be simplified as

$$PFC_i = E_i \wedge \overline{(E_{i+1} \vee \cdots \vee E_p)}.$$

5.3 Phase Algebra

Let $\overline{A} = 1$ mean that component A has failed. Then $A = 0$ says that component A has failed and $A = 1$ means that component A is operational. Using this notation, for the system described in Figure 2 the following Boolean expression describe the failure combinations for phases X, Y, and Z.

$E_X = \overline{A} + \overline{B} + \overline{C}$

$E_Y = \overline{A} + \overline{B}\,\overline{C}$

$E_Z = \overline{A}\,\overline{B}\,\overline{C}$

It should be noted that in the expression for PFC_i, event $\overline{A}$ denotes the failure of component A in phase i only. Thus for each phase, we need to define a separate symbol for each component. This is very similar to Esary and Ziehms notation where they have a separate symbol denoting failure of a component in each

phase. Let $A_i = 1$ denote the event that component A is operational during the interval from the start of the mission until the end of phase i. This automatically implies that the component is operational during all earlier phases as well. With this addition, the Boolean expressions for phases X, Y, and Z used in system phase i are denoted by E_{iX}, E_{iY}, and E_{iZ}, respectively, and are given by the following.

$$E_{iX} = \overline{A_i} + \overline{B_i} + \overline{C_i}$$

$$E_{iY} = \overline{A_i} + \overline{B_i}\ \overline{C_i}$$

$$E_{iZ} = \overline{A_i}\ \overline{B_i}\ \overline{C_i}$$

When the expression for PFC_i is simplified, we need to merge different combinations of such terms which could be a little tricky and need special treatment. Let i and j be two phases and let $i < j$. The following rules should be used to simplify the logic expressions.

$$\begin{array}{lcllcl} A_i\ A_j & \rightarrow & A_j & \qquad \overline{A_i} + \overline{A_j} & \rightarrow & \overline{A_j} \\ \overline{A_i}\ \overline{A_j} & \rightarrow & \overline{A_i} & \qquad A_i + A_j & \rightarrow & A_i \\ \overline{A_i}\ A_j & \rightarrow & 0 & \qquad A_i + \overline{A_j} & \rightarrow & 1 \end{array} \tag{2}$$

$A_i\overline{A_j}$ and $\overline{A_i} + A_j$ do not simplify any further. What the first combination means is that component A is operational until the end of phase i and then fails sometime between the end of phase i and end of phase j. The second term has no physical meaning. Also, if a component fails during a phase and then it is required to be operational during a later phase, then the two events cannot be satisfied at the same time. That is why $\overline{A_i}A_j \rightarrow 0$ holds.

The correctness of these relations can be verified by considering the following. Let $a_i = 1$ denote that the component A is operational during phase i only. Then $A_i = a_1 a_2 \cdots a_i$ and $A_j = a_1 a_2 \cdots a_j$. Now by substituting these values on both sides of each of these relations, we can verify that Relations 2 hold.

5.4 System Unreliability

Using the phase algebra, the system unreliability can be computed as follows. First compute all the PFC_i's for all phases. Then the system unreliability is given by

$$UR = P(E_p) + \sum_{i=1}^{p-1} P(PFC_i) \tag{3}$$

where $P(E_p)$ is the probability of failure evaluated using the fault tree, E_p of phase p (the last phase) using the failure distribution function calculated for each component as described in Section 3. $P(PFC_i)$ is the probability of phase failure combinations for phase i. To calculate PFC_i's, we will require probability of events such as a component remains operational during all phases starting from 1 to i, or a component remains operational during phase 1 to phase k and then fails during phase $k+1$ to phase i for some k. Such probabilities can also be calculated using the techniques defined in Section 3.

5.5 Example

In this section, we demonstrate our technique using the example described in Figure 2. This system has three components and we describe three phases, X, Y, and Z. To show the difference, we will consider all the six permutations of three phases. The failure combinations of three phases are defined by E_X, E_Y, and E_Z above.

Now we discuss each of the six permutations separately.

Permutation X Y Z. In this case first phase is phase X, followed by phase Y, that is followed by phase Z. So the PFC_i functions are obtained as follows.

$$\begin{aligned} PFC_1 &= (E_{1X} \cdot \overline{E_{2Y}}) \cdot \overline{E_{3Z}} \\ &\cdot = ((\overline{A_1} + \overline{B_1} + \overline{C_1}).(\overline{\overline{A_2} + \overline{B_2}\ \overline{C_2}})).(\overline{\overline{A_3}\ \overline{B_3}\ \overline{C_3}}) \\ &= A_3B_2\overline{C_1} + A_3\overline{B_1}C_2 + A_2B_3\overline{C_1} + A_2\overline{B_1}C_3 \end{aligned} \tag{4}$$

$$\begin{aligned} PFC_2 &= E_Y.\overline{E_Z} \\ &= (\overline{A_2} + \overline{B_2}\ \overline{C_2}).(\overline{\overline{A_3}\ \overline{B_3}\ \overline{C_3}}) \\ &= A_3\overline{B_2}\ \overline{C_2} + \overline{A_2}B_3 + \overline{A_2}C_3 \end{aligned}$$

Then the system unreliability is given by

$$\begin{aligned} UR_{XYZ} &= P(E_{3Z}) + P(PFC_1) + P(PFC_2) \\ &\textit{where} \\ P(E_{3Z}) &= P(\overline{A_3}) \cdot P(\overline{B_3}) \cdot P(\overline{C_3}) \\ P(PFC_1) &= P(A_3B_2\overline{C_1} + A_3\overline{B_1}C_2 + A_2B_3\overline{C_1} + A_2\overline{B_1}C_3) \\ &= P(A_3B_2\overline{C_1}) + P((A_3\overline{B_1}C_2 + A_2B_3\overline{C_1} + A_2\overline{B_1}C_3) \cdot (\overline{A_3} + \overline{B_2} + C_1)) \\ &= P(A_3B_2\overline{C_1}) + P(A_3\overline{B_1}C_2 + A_2\overline{A_3}B_3\overline{C_1} + A_2\overline{B_1}C_3) \\ &= P(A_3B_2\overline{C_1}) + P(A_3\overline{B_1}C_2) + P((A_2\overline{A_3}B_3\overline{C_1} + A_2\overline{B_1}C_3) \cdot (\overline{A_3} + B_1 + \overline{C_2})) \\ &= P(A_3B_2\overline{C_1}) + P(A_3\overline{B_1}C_2) + P(A_2\overline{A_3}B_3\overline{C_1} + A_2\overline{A_3}\ \overline{B_1}C_3) \\ &= P(A_3B_2\overline{C_1}) + P(A_3\overline{B_1}C_2) + P(A_2\overline{A_3}B_3\overline{C_1}) + P((A_2\overline{A_3}\ \overline{B_1}C_3) \cdot (\overline{A_2} + A_3 + \overline{B_3} + C_1)) \\ &= P(A_3B_2\overline{C_1}) + P(A_3\overline{B_1}C_2) + P(A_2\overline{A_3}B_3\overline{C_1}) + P(A_2\overline{A_3}\ \overline{B_1}C_3) \\ &\textit{and} \\ P(PFC_2) &= P(A_3\overline{B_2}\ \overline{C_2} + \overline{A_2}B_3 + \overline{A_2}C_3) \\ &= P(A_3\overline{B_2}\ \overline{C_2}) + P((\overline{A_2}B_3 + \overline{A_2}C_3) \cdot (\overline{A_3} + B_2 + C_2)) \\ &= P(A_3\overline{B_2}\ \overline{C_2}) + P(\overline{A_2}B_3 + \overline{A_2}C_3) \\ &= P(A_3\overline{B_2}\ \overline{C_2}) + P(\overline{A_2}B_3) + P((\overline{A_2}C_3) \cdot (A_2 + \overline{B_3})) \\ &= P(A_3\overline{B_2}\ \overline{C_2}) + P(\overline{A_2}B_3) + P(\overline{A_2}B_3C_3) \end{aligned} \tag{5}$$

It is easy to compute the probability of failure in phase 3 using the failure distributions for individual components. A fault tree solver such as SHARPE [2] can be used to compute that. Similarly, the probability of expressions in Equation 4 can be evaluated after simplifying the expressions as a sum of disjoint products using algorithm such as the one described in [12] and depicted in 5.

Permutation X Z Y. In this case first phase is phase X, followed by phase Z, that is followed by phase Y. Without going into details, the PFC_i functions are computed as follows.

$$PFC_1 = A_3B_3\overline{C_1} + A_3\overline{B_1}C_3$$

and

$$PFC_2 = \phi$$

The last phase in this case is phase Y. The system unreliability can be computed using

$$\begin{aligned} UR_{XZY} &= P(E_{3Y}) + P(PFC_1) + P(PFC_2) \\ &= P(\overline{A_3}) + P(A_3\overline{B_3}\ \overline{C_3}) + P(A_3B_3\overline{C_1}) + P(A_3\overline{B_1}C_3). \end{aligned}$$

Permutation Y X Z. For this case, the PFC_i functions are computed as follows.

$$PFC_1 = \phi$$

and

$$PFC_2 = A_3(\overline{B_2} + \overline{C_2}) + B_3(\overline{A_2} + \overline{C_2}) + C_3(\overline{A_2} + \overline{B_2})$$

The last phase in this case is phase Z. The system unreliability can be computed using the following. (We are omitting details of simplification.)

$$\begin{aligned} UR_{YXZ} &= P(E_{3Z}) + P(PFC_1) + P(PFC_2) \\ &= P(\overline{A_3}) \cdot P(\overline{B_3}) \cdot P(\overline{C_3}) + P(A_3\overline{B_2}) + P(A_3B_2\overline{C_2}) + P(\overline{A_2}B_3) \\ &+P(\overline{A_2}\ \overline{B_3}C_3) + P(A_2\overline{A_3}B_3\overline{C_2}) + P(A_2\overline{A_3}\ \overline{B_2}C_3) \end{aligned}$$

Permutation Y Z X. For this case, the PFC_i functions are computed as follows.

$$PFC_1 = \phi$$

and

$$PFC_2 = \phi$$

The last phase in this case is phase X. The system unreliability can be computed using the following.

$$\begin{aligned} UR_{YZX} &= P(E_{3X}) + P(PFC_1) + P(PFC_2) \\ &= P(\overline{A_3}) + P(\overline{B_3}) + P(\overline{C_3}) \end{aligned}$$

Permutation Z X Y. For this case, the PFC_i functions are computed as follows.

$$PFC_1 = \phi$$

and

$$PFC_2 = A_3B_3\overline{C_2} + A_3\overline{B_2}C_3$$

The last phase in this case is phase Y. The system unreliability can be computed using the following.

$$\begin{aligned} UR_{ZXY} &= P(E_{3Y}) + P(PFC_1) + P(PFC_2) \\ &= P(\overline{A_3}) + P(A_3\overline{B_3}\ \overline{C_3}) + P(A_3B_3\overline{C_2}) + P(A_3\overline{B_2}C_3) \end{aligned}$$

Permutation Z Y X. For this case, the PFC_i functions are computed as follows.

$$PFC_1 = \phi$$

and

$$PFC_2 = \phi$$

The last phase in this case is phase X. The system unreliability can be computed using the following.

$$\begin{aligned} UR_{YZX} &= P(E_{3X}) + P(PFC_1) + P(PFC_2) \\ &= P(\overline{A_3}) + P(\overline{B_3}) + P(\overline{C_3}) \end{aligned}$$

5.6 Exact Solution Using Markov Chain

The same three component system can be analyzed using Markov Chain for the six permutations. There are eight possible states in each phase as depicted in Figure 3. Using the same notation for the names of states, i.e., state 101 represents that components A and C are operational and component B has failed, we can derive expressions for states occupacncy probabilities (SOPs) at the end of each phase. Depending on the success criteria, for the failure states in phase p, the initial state occupancy probability for the same state in phase $p+1$ is zero.

Let $P_{p(s)}$ denote the SOP for phase p of state s where $s \in \{000, 001, 010, 011, 100, 101, 110, 111\}$ and $p = 1$, 2, and 3. Again, let T_p denote the phase duration for phase p and let CT_p denote the sum of durations of first p phases. Let λ_{A_p}, λ_{B_p}, and λ_{C_p} denote the failure rates of components A, B, and C, respectively, in phase p. Using these notations, the SOPs for phase p can be derived using the SOPs for phase $p-1$ and are given in Equation 6.

Using the relationship in Equation 6, we can compute the SOPs for operational states for each phase. The unreliability at the end of each phase is given by 1 - sum of SOPs of operational states in that phase. At the end of that phase, SOP for the failure states in that phase can be set to zero as this probability mass is not carried forward to the next phase to success states. For example, for the case of permutation X Y Z,

initially $P_{0(s)} = 0.0$ for all states where $s \neq 111$ and $P_{0(111)} = 1.0$. Using these values and the success criteria for phase X, at the end of phase X, we assign $P_{1(s)}(CT_1) = 0.0$ for all states where $s \neq 111$ and $P_{1(111)}(CT_1)$ is calculated Equation 6. Using, these values and the success criteria of phase Y, we can compute SOPs for phase 2. At the beginning of phase 3, we assign $P_{2(s)}(CT_2) = 0.0$ where $s \in \{000, 001, 010, 011, 100\}$ and compute $P_{2(s)}(CT_2) = 0.0$ where $s \in \{101, 110, 111\}$ using relations defined in Equation 6. Finally, using these results of phase 2, we can calculate $P_{3(s)}(CT_3)$ where $s \in \{001, 010, 011, 100, 101, 110, 111\}$.

Sometimes a backward or need-based computation may be more useful. For example, for permutation Z Y X, we only need to calculate $P_{3(111)}(CT_3)$ which requires only $P_{2(111)}(CT_2)$. This, in turn, requires computation of $P_{1(111)}(CT_1)$ which can be calculated using $P_{0(111)}(CT_0) = 1.0$. Finally, the unreliability for the 3 phase system is $1 - P_{3(111)}(CT_3)$. However, intermediate unreliabiities at the end of phases 1 and 2 may require more computation.

$$
\begin{aligned}
P_{p(111)}(CT_{p-1}+t) &= P_{p-1(111)}(CT_{p-1})e^{-\lambda_{A_p}t}\ e^{-\lambda_{B_p}t}\ e^{-\lambda_{C_p}t} \\
P_{p(110)}(CT_{p-1}+t) &= P_{p-1(111)}(CT_{p-1})e^{-\lambda_{A_p}t}\ e^{-\lambda_{B_p}t}\ (1-e^{-\lambda_{C_p}t}) + P_{p-1(110)}(CT_{p-1})e^{-\lambda_{A_p}t}\ e^{-\lambda_{B_p}t} \\
P_{p(101)}(CT_{p-1}+t) &= P_{p-1(111)}(CT_{p-1})e^{-\lambda_{A_p}t}\ (1-e^{-\lambda_{B_p}t})\ e^{-\lambda_{C_p}t} + P_{p-1(101)}(CT_{p-1})e^{-\lambda_{A_p}t}\ e^{-\lambda_{C_p}t} \\
P_{p(011)}(CT_{p-1}+t) &= P_{p-1(111)}(CT_{p-1})(1-e^{-\lambda_{A_p}t})\ e^{-\lambda_{B_p}t}\ e^{-\lambda_{C_p}t} + P_{p-1(011)}(CT_{p-1})e^{-\lambda_{B_p}t}\ e^{-\lambda_{C_p}t} \\
P_{p(100)}(CT_{p-1}+t) &= P_{p-1(111)}(CT_{p-1})e^{-\lambda_{A_p}t}\ (1-e^{-\lambda_{B_p}t})\ (1-e^{-\lambda_{C_p}t}) + P_{p-1(100)}(CT_{p-1})e^{-\lambda_{A_p}t} \\
&\quad + P_{p-1(110)}(CT_{p-1})e^{-\lambda_{A_p}t}\ (1-e^{-\lambda_{B_p}t}) + P_{p-1(101)}(CT_{p-1})e^{-\lambda_{A_p}t}\ (1-e^{-\lambda_{C_p}t}) \\
P_{p(010)}(CT_{p-1}+t) &= P_{p-1(111)}(CT_{p-1})(1-e^{-\lambda_{A_p}t})\ e^{-\lambda_{B_p}t}\ (1-e^{-\lambda_{C_p}t}) + P_{p-1(010)}(CT_{p-1})e^{-\lambda_{B_p}t} \\
&\quad + P_{p-1(110)}(CT_{p-1})(1-e^{-\lambda_{A_p}t})\ e^{-\lambda_{B_p}t} + P_{p-1(011)}(CT_{p-1})e^{-\lambda_{B_p}t}\ (1-e^{-\lambda_{C_p}t}) \\
P_{p(001)}(CT_{p-1}+t) &= P_{p-1(111)}(CT_{p-1})(1-e^{-\lambda_{A_p}t})\ (1-e^{-\lambda_{B_p}t})\ e^{-\lambda_{C_p}t} + P_{p-1(001)}(CT_{p-1})e^{-\lambda_{C_p}t} \\
&\quad + P_{p-1(101)}(CT_{p-1})(1-e^{-\lambda_{A_p}t})\ e^{-\lambda_{C_p}t} + P_{p-1(011)}(CT_{p-1})(1-e^{-\lambda_{B_p}t})\ e^{-\lambda_{C_p}t}
\end{aligned}
\tag{6}
$$

5.7 Comparison with Other Techniques

We analyze the above six scenarios using the technique discussed in this paper, Esary and Ziehms approach, analytic solution of Markov chains, phased-mission approach of [10] and [9], and the phased-mission approach of [8]. We assume that the durations of all the three phases are 10 hours each and the failure rate of each of the component is 0.0001/hour. Thus the input data do not skew results in any direction as all components are similar and all phases are similar. The results are only affected by the sequencing of phases and system success criteria.

We obtain the results shown in Tables 1 and 2. The results for the six permutations of phases X, Y, and Z, are obtained (and listed) at the end of each phase. When the worst case criteria is applied, that is a failed state in one phase is considered as failed state in all subsequent phases, the results for unreliability can be very high.

Table 1: Unreliability of Phased-Mission System (Accurate Analysis)

Permute	X Y Z	X Z Y	Y X Z	Y Z X	Z X Y	Z Y X
Phase 1	0.002995504	0.002995504	0.001000498	0.001000498	0.000000001	0.000000001
Phase 2	0.003993006	0.002995505	0.005982036	0.001000502	0.005982036	0.002001985
Phase 3	0.003993009	0.004991493	0.005982037	0.008959621	0.006976549	0.008959621

Table 2: Unreliability of Phased-Mission System (Worst Case Scenario)

Permute	X Y Z	X Z Y	Y X Z	Y Z X	Z X Y	Z Y X
Phase 1	0.002995504	0.002995504	0.001000498	0.001000498	0.000000001	0.000000001
Phase 2	0.005982036	0.005982036	0.005982036	0.002001985	0.005982036	0.002001985
Phase 3	0.008959621	0.008959621	0.008959621	0.008959621	0.008959621	0.008959621

The important thing to observe here is that when we allow failure combinations (failure states in Markov chains) to become operational combinations (up states in Markov chains) in a later phase, then the overall unreliability of the system can be substantially lower, as is the case in the last column. For example, in a spacecraft, launch is the most important activity. After that, all launch related activities or components which could have caused failure during launch is not going to make any difference any more. Thus those failure combinations are operational combinations for the rest of the mission.

To further explore the impact of phase configurations and durations of phases, we varied the phase durations. In the first variation, we assume that the first phase is always of 1 hour duration, the second phase is of 10 hour duration, and the third phase is of 100 hour duration irrespective of the types of phase configurations, X, Y, or Z, used during these phases. The results for this variation for the two cases are shown in Tables 3 and 4, respectively. In another variation, we assume that the phase X is always of 1 hour duration, phase Y is always of 10 hours duration, and phase Z is always of 100 hours duration irrespective of where in the mission these phase configurations are used. The results are given in Tables 5 and 6, respectively. In this case, the results differ by more than an order of magnitude depending on the ordering of the phases. If the stringest success criteria is during the beginning of phases, then phased-mission analysis is more meaningful.

It should be noted that the techniques in [10], [8], and [9] are capable of handling the more general case of repairable systems while the technique discussed by Esary and Ziehms as well as the one presented in this paper are both restricted to the cases of non-repairable systems. The technique in [9] is most general but most expensive in computation time and in this case will yield the same result as in [10] because both of these make no approximations.

Table 3: Unreliability with 1, 10, and 100 hours phases (Accurate Analysis)

Permute	X Y Z	X Z Y	Y X Z	Y Z X	Z X Y	Z Y X
Phase 1	0.000299955	0.000299955	0.000100005	0.000100005	0.000000000	0.000000000
Phase 2	0.001300153	0.000299956	0.003294561	0.000100006	0.003294561	0.001100603
Phase 3	0.001301332	0.011354728	0.003295543	0.032751658	0.013309644	0.032751658

Table 4: Unreliability with 1, 10, and 100 hours phases (Worst Case Scenario)

Permute	X Y Z	X Z Y	Y X Z	Y Z X	Z X Y	Z Y X
Phase 1	0.000299955	0.000299955	0.000100005	0.000100005	0.000000000	0.000000000
Phase 2	0.003294561	0.003294561	0.003294561	0.000200020	0.003294561	0.001100603
Phase 3	0.032751658	0.032751658	0.032751658	0.032751658	0.032751658	0.032751658

6 Conclusions

We have presented a technique to analyze phased-mission systems using fault trees. This technique yields accurate results and is simpler in concept and computation. For this purpose, we develop a phase algebra that allows us to efficiently compute the probability of all possible combinations contributing to failure in phased-mission systems during individual phases. This technique will be very useful for a large class of systems where the system behavior can be described using fault trees.

References

[1] K. Trivedi, J. Dugan, R. Geist, M. Smotherman, B. Rothman, M. Boyd, S. Bavuso, "HARP: The Hybrid Automated Reliability Predictor - Introduction and guide for the users," Dept. of Computer Science, Duke University, Durham NC 27706, March 1986.

[2] R. Sahner, K. Trivedi, "Reliability Modeling using SHARPE", *IEEE Trans. Reliability*, vol. R-36, June 1987, pp. 186-193.

[3] A. Conway and A. Goyal, "Monte Carlo Simulation of Computer System Availability/Reliability Models", *FTCS-16*, June 1986.

[4] R. Butler, "The SURE Reliability Analysis Program", *AIAA Guidance, Navigation, and Control Conference*, Williamsburg, Virginia, August 18-20, 1986.

[5] J. D. Esary and H. Ziehms, "Reliability Analysis of Phased Missions," *Proc. of the Conf. on Reliability and Fault Tree Analysis*, SIAM, 1975, pp. 213-236.

Table 5: Unreliability with $T_X = 1$, $T_Y = 10$, and $T_Z = 100$ hours phases (Accurate Analysis)

Permute	X Y Z	X Z Y	Y X Z	Y Z X	Z X Y	Z Y X
Phase 1	0.000299955	0.000299955	0.001000498	0.001000498	0.000000985	0.000000985
Phase 2	0.001300153	0.000300940	0.003294561	0.001001678	0.029845556	0.011058089
Phase 3	0.001301332	0.011354728	0.003295543	0.032751658	0.030816194	0.032751658

Table 6: Unreliability with $T_X = 1$, $T_Y = 10$, and $T_Z = 100$ hours phases (Worst Case Scenario)

Permute	X Y Z	X Z Y	Y X Z	Y Z X	Z X Y	Z Y X
Phase 1	0.000299955	0.000299955	0.001000498	0.001000498	0.000000985	0.000000985
Phase 2	0.003294561	0.003294561	0.003294561	0.011058089	0.029845556	0.011058089
Phase 3	0.032751658	0.032751658	0.032751658	0.032751658	0.032751658	0.032751658

[6] A. Pedar and V.V.S. Sarma, "Phased-Mission Analysis for Evaluating the Effectiveness of Aerospace Computing-Systems," *IEEE Trans. on Rel.*, vol. R-30, No.5, Dec. 1981, pp. 429-437.

[7] M. Alam and U. Al-Saggaf, "Quantitative Reliability Evaluation of Repairable Phased-Mission Systems Using Markov Approach," *IEEE Trans. on Rel.*, vol. R-35, No.5, Dec. 1986, pp. 498-503.

[8] J.B. Dugan, "Automated Analysis of Phased-Mission Reliability," *IEEE Trans. on Rel.*, vol. R40, No. 1, April 1991, pp. 45-51.

[9] M. Smotherman and K. Zemoudeh, "A Non-Homogeneous Markov Model for Phased-Mission Reliability Analysis," *IEEE Trans. on Rel.*, vol. R-38, No.5, Dec. 1989, pp. 585-590.

[10] A. Somani, J. Ritcey, and S. Au, "Phased Mission Reliability Analysis," abstract and poster presentation in *Proc. of SIGMETRICS-90*, May 1990. A full version, "Computationally Efficient Phased-Mission Reliability Analysis for Systems with Variable Configuration," *IEEE Trans. on Rel.*, vol. R-42, Dec. 1992, pp. 504-509.

[11] A. Somani, S. Palnitkar, and T. Sharma, "Reliability Modeling of Systems with Latent failures using Markov Chains," in *Proc. of RAMS-93*, pp. 120-125.

[12] M. Veeraraghavan and K. S. Trivedi, "An Improved Algorithm for Symbolic Reliability Analysis," *IEEE Trans. on Rel.*, vol. R-40, No.3, Dec. 1991, pp. 347-358.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	November 1993	Contractor Report

4. TITLE AND SUBTITLE
PHASED-MISSION SYSTEM ANALYSIS USING BOOLEAN ALGEBRAIC METHODS

5. FUNDING NUMBERS
C NAS1-19480
WU 505-90-52-01

6. AUTHOR(S)
Arun K. Somani
Kishor S. Trivedi

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Institute for Computer Applications in Science and Engineering
Mail Stop 132C, NASA Langley Research Center
Hampton, VA 23681-0001

8. PERFORMING ORGANIZATION REPORT NUMBER
ICASE Report No. 93-74

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
NASA CR-191564
ICASE Report No. 93-74

11. SUPPLEMENTARY NOTES
Langley Technical Monitor: Michael F. Card
Final Report
Submitted to Sigmetric '94

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Unclassified–Unlimited

Subject Category 61

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

Most reliability analysis techniques and tools assume that a system is used for a mission consisting of a single phase. However, multiple phases are natural in many missions. The failure rates of components, system configuration, and success criteria may vary from phase to phase. In addition, the duration of a phase may be deterministic or random. Recently, several researchers have addressed the problem of reliability analysis of such systems using a variety of methods. We describe a new technique for phased-mission system reliability analysis based on Boolean algebraic methods. Our technique is computationally efficient and is applicable to a large class of systems for which the failure criterion in each phase can be expressed as a fault tree (or an equivalent representation). Our technique avoids state space explosion that commonly plague Markov chain-based analysis. We develop a phase algebra to account for the effects of variable configurations and success criteria from phase to phase. Our technique yields exact (as opposed to approximate) results. We demonstrate the use our technique by means of an example and present numerical results to show the effects of mission phases on the system reliability.

14. SUBJECT TERMS
Ultra-Reliable Computer System; Reliability Analysis; Phased-Mission Systems; Fault Trees; Boolean Algebraic Methods; Reconfiguration; Variable Success Criteria; Random Phase Duration

15. NUMBER OF PAGES
20

16. PRICE CODE
A03

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified		

NSN 7540-01-280-5500

Standard Form 298(Rev. 2-89)
Prescribed by ANSI Std. Z39-18
298-102

☆U.S. GOVERNMENT PRINTING OFFICE: 1993 - 528-064/86093