

Simplified Phased-Mission System Analysis for Systems with Independent Component Repairs *

Arun K. Somani

Dept. of Electrical Eng., and Dept. of Computer Sci. and Eng.

University of Washington

P O Box 352500

Seattle, WA 98195-2500

Tele: (206) 685-1602

email: {somani,arun}@{ee,cs}.washington.edu

Abstract

Accurate analysis of reliability of system requires that it accounts for all major variations in system's operation. Most reliability analyses assume that the system configuration, success criteria, and component behavior remain the same. However, multiple phases are natural. We present a new computationally efficient technique for analysis of phased-mission systems where the operational states of a system can be described by combinations of components states (such as fault trees or assertions). Moreover, individual components may be repaired, if failed, as part of system operation but repairs are independent of the system state. For repairable systems Markov analysis techniques are used but they suffer from state space explosion. That limits the size of system that can be analyzed and it is expensive in computation. We avoid the state space explosion. The phase algebra is used to account for the effects of variable configurations, repairs, and success criteria from phase to phase. Our technique yields exact (as opposed to approximate) results. We demonstrate our technique by means of several examples and present numerical results to show the effects of phases and repairs on the system reliability/availability.

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

1 Introduction

Accurate analysis of reliability of system requires that it accounts for all major variations in system's operation. Most reliability analyses assume that the system configuration, success criteria, and component behavior remain the same. However, multiple phases are natural. 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. Various techniques and tools have been developed [1]-[4] to analyze single mission system. Phased-mission system analysis also has received substantial attention by researchers [5] - [12].

Depending on the requirements during different phases, different components may be placed in or removed from service or repaired during a phase to balance the system reliability and the cost of operation. The success of a redundancy management scheme determines if a system is operational or not. The usage of subsystems may also vary from phase to phase and subsystem 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. 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 behaviors.

Sometimes the effects of individual phases may be ignored in favor of simpler analysis. For example, in case of landing gear example, if the failure rate of landing gear is very small for all phases, counting the failure of landing gear during entire flight may not affect result significantly. On the other hand, in another example, in a space mission, the first phase (launch) is the most severe and uses many components for a few minutes whose failure rates are high. Using the high failure rates and exposure time equal to the mission time for those components is guaranteed to result into useless analysis.

In approximate analysis, most of the time only conservative estimates are made yielding the worst case unreliability of the system. One adverse effect of this is that the systems may be over-designed. A more accurate analysis avoids this, in particular where there may be wide variations in the parameters and system configuration from phase to phase. If one phase experiences much more stress than others then it is necessary to account for such effects properly. Different aspects of phased-mission analysis are discussed by several researchers [5] - [12].

A phased-mission system can be analyzed accurately using Markov methods. However that suffers from state-space explosion and is expensive in time. In [12], the authors presented a methodology to analyze non-repairable phased-mission systems in which failure rates, configuration and success criteria may vary from phase to phase. Moreover, the success criteria can be specified using fault trees or an equivalent representation. A majority of systems can be represented using fault trees. They solve the system without generating a Markov chain. Phases are handled one at a time to compute the overall unreliability of the entire mission. This technique is computationally less expensive. As a result, large systems can be managed.

It is possible that during long missions, repairs are carried out on components or subsystems to increase the life of system. For example, in a long manned space mission, failed components will be repaired and must be appropriately accounted for in the analysis. The form of repair may vary. For example, a system may be completely replaced by another new system or only maintenance checks may be carried out and subsystems are repaired in the conventional sense. Markov analysis techniques can be used but, as stated earlier, may require to manage huge state space and computation time. We extend the methodology of [12] in this paper significantly by including repairs of independent components. We require that the system success criteria is dependent only on the state of individual component and as long as the success criteria is satisfied, the phase remains operational. The results of this paper allows analysis of large systems with component repairs efficiently. In the descriptions below, we will assume that a reader is generally familiar with Markov chain-based analysis. We will use it to describe certain situations but will propose a methodology which does not explicitly generate the state space.

In all of this work, phase transitions are assumed to be instantaneous and no loss or gain is assumed in the probability of any particular state in Markov chain. However, due to change in success criteria, some operational states may be seen as failure states in the next phase and are treated as latent failures for analysis. For example, if the landing gear develops a problem during cruising, the flight will continue in air but the last phase, landing, may not be successful. Thus the landing gear failure is latent. If the failed landing gear can be repaired during the flight, then the effect can be accounted for in the analysis.

We present some related work in the next section. Then we describe some concepts which we will use throughout the paper. Following that we present handling of repairable systems and our methodology to manage computation efficiently. We present a few examples and demonstrate the effectiveness of our work. In all cases, the results are compared with EHARP [10] results which compute unreliability of phased mission system correctly as it follows state-to-state mapping from phase to phase.

2 Related Work

Esary and Ziehms [5] discuss analysis of multiple configuration systems during different phases of a mission using reliability block diagram (RBD). For phase p , each component is represented by a series of a blocks, one corresponding to each phase starting with phase 1 to phase p . All phase RBDs are connected in series and solution of this RBD correctly predicts the reliability of the three phase system. This results in a large RBD and failure of components cannot be accounted for. Pedar and Sarma [6] enhanced this technique to systematically cancel out the common events in earlier phases which are accounted for in later phases in the RBDs. We will use Esary and Ziehms's representation for components in various phases for analysis but perform the computation differently.

Alam and Al-Saggaf [7] use Markov chain and Smotherman et. al. [9] use a non-homogeneous Markov model to include phase changes in the model. The Markov chain in both cases can be very huge. It should be pointed out that the latter technique allows the most accurate analysis if phase changes are not smooth. However, this requires large amount of storage and computation time to solve a system, thus limiting the type of system 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. Instead of generating and solving an overall Markov chain, they advocate generating and solving separate Markov chains for individual phases. The variation in success criteria and change in system configuration from phase to phase are accommodated by providing an efficient mapping procedure at the transition time from one phase to another. While analyzing a phase, only the states relevant to that phase, are considered. Thus each individual Markov chain is much smaller.

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. However, the failure criteria is also the union of all phases failure criteria as any failed state in any phase is considered failed state for the whole system. Thus, the scheme is only applicable is the success criteria does not change over the phases.

3 Distribution Functions with Mass at Origin

As in [12], we will use the concept of cumulative distribution functions with a mass at the origin in our work. 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 constant 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 first 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.

3.1 Component Model with Repairs

The model described above can be extended to include repair for a component. Let X be a component whose failure and repair rates in phase p are denoted by λ_{Xp} and μ_{Xp} , respectively. Failure and repair times are assumed to follow exponential distribution. We define

$$\alpha_{Xp}(t) = e^{-(\lambda_{Xp} + \mu_{Xp}) * t} \quad \text{and} \quad \beta_{Xp} = \frac{\mu_{Xp}}{\mu_{Xp} + \lambda_{Xp}} \quad (1)$$

where t is the time after the system entered the phase p . We can compute probabilities of component X being operational (up) or not-operational (failed) by solving a two state Markov chain for the component. At the beginning of a phase a component may be in an operational or failed state. With either of the initial states, the component may be operational or failed at the end of the phase due to failure and repairs involved during that phase. To compute the probabilities for a component to be operational or failed at the end of the phase, we need to compute the probabilities of all the four possible cases.

We will follow a 4 character suffix with probabilities. The first character is the name of the component (i.e. X, Y). The second character is u for up or f for failed and is associated with the starting state of that component in a phase. The third character is u or f as earlier. It can also be e if it refers to probability at the end of a phase or a b if it refers to the probability at the beginning of a phase. The fourth character p is for phase number. The first and the fourth characters will change with components or phase number we are dealing with. If it is given that the component X is up, then the probabilities that it will remain up or failed after time t has elapsed in phase p are given by

$$p_{Xuup}(t) = \alpha_{Xp}(t) + \beta_{Xp} * (1 - \alpha_{Xp}(t)) \quad (2)$$

and

$$p_{Xufp} = (1 - \alpha_{Xp}(t)) * (1 - \beta_{Xp}). \quad (3)$$

Similarly if it is given that component X is failed, then the probabilities that it will remain up or failed are given by

$$p_{Xfup} = \beta_{Xp} * (1 - \alpha_{Xp}(t)) \quad (4)$$

and

$$p_{Xffp} = 1 - \beta_{Xp} * (1 - \alpha_{Xp}(t)). \quad (5)$$

If the probabilities that component X is initially up and failed at the beginning of the phase p are p_{Xubp} and p_{Xfbp} , respectively, then the probabilities that the component is up or failed after time t has elapsed in phase p are given by

$$P_{Xuep}(t) = p_{Xubp} * p_{Xuup}(t) + p_{Xfbp} * p_{Xfup}(t) \quad (6)$$

and

$$P_{Xfep}(t) = p_{Xubp} * p_{Xufp}(t) + p_{Xfbp} * p_{Xffp}(t). \quad (7)$$

The overall operational and failed state probabilities for a component can be evaluated at the end of phase p by substituting $t = T_p$ in the the above expressions. They include the mass at the origin (the initial up or failed state probabilities). T_p is the duration of phase p . For example, suppose for a component X in phase 1, if $\mu_{X1} = 9 * \lambda_{X1}$, $T_1 = 10$ hrs, and μ_{X1} and λ_{X1} are chosen so that $\alpha_{X1}(10) = 0.9$. $\beta_{X1} = 0.9$. Then, $p_{Xuu1} = 0.99$, $p_{Xuf1} = 0.01$, $p_{Xfu1} = 0.09$, and $p_{Xff1} = 0.91$. If $p_{Xub1} = 1.0$ and $p_{Xfb1} = 0.0$, then $p_{Xue1} = 0.99$ and $p_{Xfe1} = 0.01$. If, on the other hand, $p_{Xub1} = 0.99$ and $p_{Xfb1} = 0.01$, then $p_{Xue1} = 0.99 * 0.99 + 0.01 * 0.09 = 0.981$ and $p_{Xfe1} = 0.99 * 0.01 + 0.01 * 0.91 = 0.019$.

4 Phased-Mission and Component Repairs

In analysis of reliable system when a system enters a failure state during a phase, the entire mission is considered to have failed. So the next phase only begins, if the system remains operational during all previous phases. If the components are not repaired, the success or failure of system depends on the cumulative operational probabilities and success criteria defined by the combinations of states of operational components. In such cases, as shown in [10]-[12], one can compute the success probability of the whole mission.

Notice that a system state may be considered as a failed state in phase p but may be a success state in the next phase due to a less stringent success criteria. This is acceptable behavior even in reliable systems. In such cases, all state occupation probabilities (SOPs) accumulated in such states up to only phase p are considered to be contributing towards failure of mission. Thereafter they are considered as part of success. This is key to correct analysis of a phased-mission system and is implemented in EHARP.

In certain situations, however, it is possible to design systems that include repairs to keep reliability high. For example, in a long mision, to improve reliability and performance, it may be advisable and necessary to carry out repairs on system during operation of system. Since in different phases success criterias vary, all of the components may not be used in all phases. When certain components are not required for the system operation, they may be repaired and employed again in the following phases. The repairs are to remain in ready state for future phases. In phases when repairs are carried out, the system status is not affected by the components under repairs. In Markov chain representation this implies that the repair transitions are from failed states to failed states or operation states to operation status. In such cases, we can compute reliability more efficiently using the approach of this paper.

For example consider two components, A and B, system which are used alternately in two consecutive phases.

Both components can fail in either phase but only the component not in use in a phase only undergoes repairs in that phase. The system operational and failed states for the two phases are shown in Figure 1.

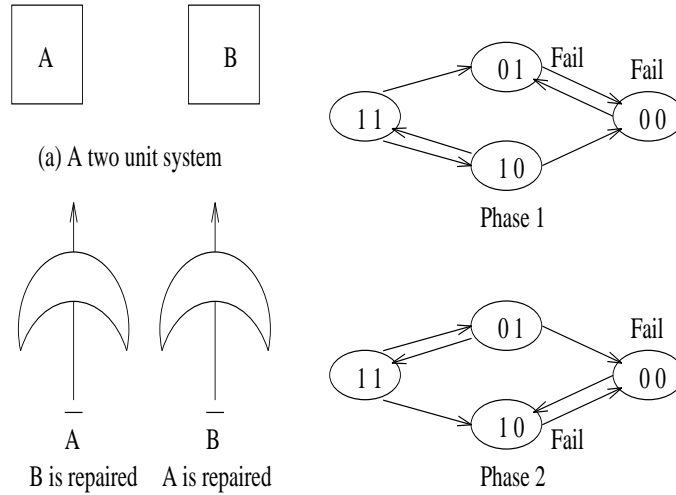


Figure 1: A two component system and its failed states

In a repairable system, it is also possible that the system may enter from a failed state to a success state within the same phase. Since the success criteria is specified using combinatorial methods, this will happen if the system up or failed state depends on a component which is also being repaired in that phase. In such cases, use of combinatorial methods only will not allow us to pay us attention to the fact the system may transit through the failed states. One important consideration here is that must such transitions be allowed in the same phase? Strictly speaking, for critical operation system, once a system failure has occurred, it is catastrophic and must be treated as such. This is, therefore, obviously not allowed for reliable system as they are considered failed once the system enters a failed state. In that case, the technique of this paper cannot be applied as the system does not remain symmetric. Such systems can only be solved using the techniques described in [7, 9, 10] and the tools such as EHARP.

There are many other scenarios where the techniques developed in this paper will apply. In this paper we are assuming that component repairs are independent of system states and are carried out based on the component states only, the success criteria may be such that this does not impact the results. If only those components are repaired that are not participating in the operation of a system in that phase then the success criteria automatically satisfies the requirement for correct analysis. This is the case in the example of Figure 1. This is because the up or failed state of such components would not affect the analysis as they do not affect the success criteria. Alternatively, if the approach for success is that “all is well if the end is well,” then also this analysis can be used. What we mean by this is that if it is the system state at the end of a phase that counts and transient states during the operation do not matter (or do not matter “much”), then this technique can be used.

Another question that arises is that can one start the next phase or not in a state where the system is considered failed. For reliability analysis, the obvious answer is no as the system has already failed. But in some analysis, like performability or availability, this is obviously acceptable. Thus handling of such states depends on the system definition. This is open to interpretation. For availability and performability analysis, if a particular phase may fail in a particular combination, that combination may be considered further as the system may recover from it due to repairs. In such cases, it is possible, that the next phase can begin, even if the system is in a failed state since it is possible that the system is brought back up in an operational state. So, in essence we may be more interested in the availability of a system during a particular phases and not reliability according to definition of reliability. The availability then can be used to compute the performability of the system. This analysis is beyond the scope of this paper and is subject of our further research.

4.1 Examples Used in the Paper

To describe and show the effectiveness of the work here, we will use the following three examples.

Example 1. Our first example is the the one described earlier of a two components A and B, system that can be represented using four states in a Markov chain as shown in Figure 1. One component is repaired while the other is used for the system operation. Thus failure and success of system depends on the component being used. This may correspond to a factory floor where two machines are alternately used while other goes through its repair (or maintenance) cycle and is repaired as needed to bring it up to the fully operational state. We will consider a four phased system with different parameters and phase durations.

Example 2. The second example is of a slightly bigger system where we have more scope to show changes in system configuration that lead to system failure and success and finer points of the complexity involved in analysis. This system consists of three component, A, B, and C. One of these components may be repaired in a phase while the other two are used in a phase in some combinations. The system remains operational as long as the specified success criteria is satisfied. The success criteria for each of the three phases is expressed using fault trees. Each time we use two components and depending on the requirements we may require both or any one of them operational. The failure rates of three components are λ_a , λ_b , and λ_c , respectively, and these are defined for each phase separately. The repair rates for these parameters are μ_a , μ_b , and μ_c , respectively. Two particular configuration using two out of the three component are shown in Figure 2a.

A Markov chain for a three component system with all repair arcs is also shown in Figure 2b. 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

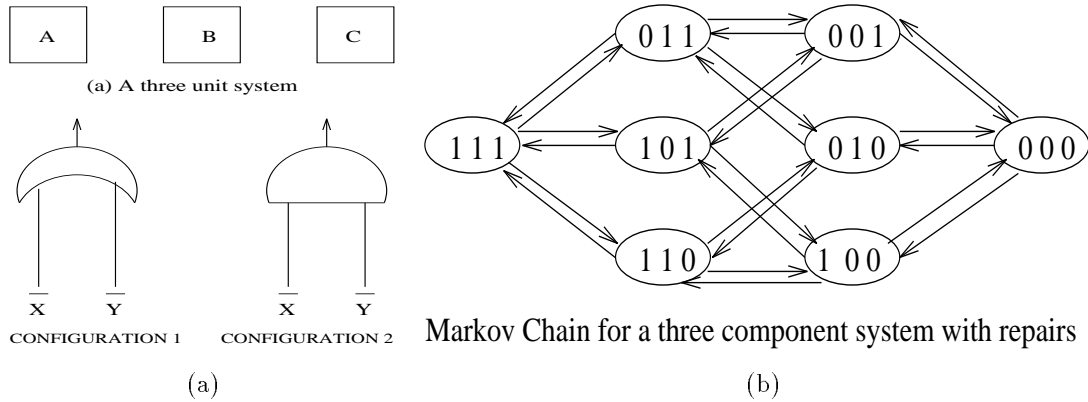


Figure 2: (a) Two configuration of a three component system and (b) the Markov chain with all failure and repair arcs.

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 or repair rate of the component that is repaired. For example, a transition from state (011) to state (010) has a transition rate of λ_c . States marked F are failure states. Similarly, a transition from state (010) to state (011) has a transition rate of μ_c .

Depending on success criteria and system parameters, only some of these states will be success states in each phase. Some of the arcs may have 0 rate associated with them or they may not exist. For example, if a repair is not active, the corresponding arc may be dropped. We will use several combination of two possible success criterias in a three phase system. In each of these cases, one of the components will not be used in each phase and will be repaired. The component parameters and phase duration may vary.

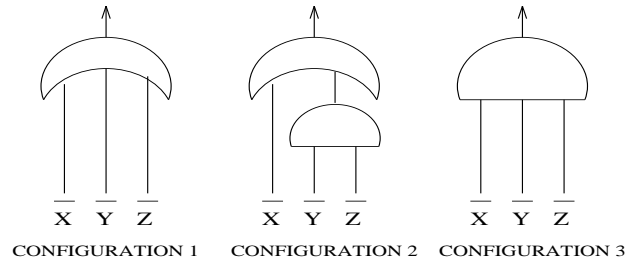


Figure 3: (a) Three configuration of a three component system.

Example 3. For our third example, we will use “all is well if the end is well approach.” We will use the same three component system of Example 2 but will use all three components in each phase. The three phase configurations to be used are shown in Figure 3. The components are also repaired in each phase. As long as a

phase terminates satisfying the success criteria. We will compare the results with the case when repair arcs are not allowed from the failed state (analysis performed using EHARP) and to notice the inaccuracies incurred in computation.

5 Phased-Mission Analysis

Suppose we are given the failure, and repair rates for each component for each phase and the success criteria for each phase. The component failure and repair rates may be phase dependent. We assume that the phase durations are deterministic.

To account for phase-dependent failure and repair rates, we use the component model for failure and success distribution with mass at origin for each component as described in Section 3.1. We compute the distribution of failure for each component for each phase using the initial (beginning of that phase) up and failed probabilities and failure and repair rates for that phase. The failure distribution function is described in Equation 7. In there, time t is measured from the beginning of phase p so that $0 \leq t \leq T_p$. T_p represents the duration of phase p . This expression is in recursive form and can be further simplified by substituting $P_{Xubp} = P_{Xue(p-1)}(T_{p-1})$ (the final values for phase $p-1$ as the initial values for phase p). But we prefer to leave the expressions for each phase as they are in the recursive form as we need individual phase components in our computation to combine the results for all phases together.

Notice that a component may be up or failed in any phase with the distributions described in Equations 6 and 7 irrespective of its status in the previous phase due to failure and repairs of that component in that phase. This is in contrast to non-repairable system where a component can be up only if it is up at the beginning of the phase.

If the failure and repair rates are age-dependent, then one would have to consider time as a global parameters, i.e., time starts with the beginning of a mission and phase p starts at time $CT_{p-1} = \sum_{i=1}^{p-1} T_i$ and finishes at $CT_p = \sum_{i=1}^p T_i$. The probabilities P_{xuu_p} , P_{xuf_p} , P_{xfu_p} , and P_{xff_p} are calculated using a single component model where both failure and repair rates are function of time. The resulting component behavior is represented using a more complicated non-homogeneous Markov chain for which appropriate differential equations can be easily developed. However, solution of these equations does not have a closed form solution for general $\mu(t)$ and $\lambda(t)$ [14]. In specific cases when $\mu_{X_p}(t) = 0$ and only failure rate $\lambda_{X_p}(t)$ is a function of time, we can compute $p_{xfu_p} = 0.0$, $p_{xff_p} = 1.0$, $p_{xuu_p} = 1 - e^{-\int_{CT_{p-1}}^{CT_p} \lambda_{X_p}(\tau) d\tau}$ and $p_{Xuu_p} = e^{-\int_{CT_{p-1}}^{CT_p} \lambda_{X_p}(\tau) d\tau}$. The rest of the computation remains the same.

5.1 Management of Phase-Dependent Success Criteria

The success criteria in different phases may be different for a variety of reasons including (i) not all components are used in all phases, (ii) the expected performance out of individual components may be different in different phases, (iii) individual subsystems may be dropped or included in the system, (iv) the dropped (not used) subsystem may be repaired, and (v) additional redundancy may be provided or redundancy levels may be reduced for certain tasks.

Due to a change in success criteria and repairs, 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. The following five scenarios arise in computation at the time of phase transition from phase p to phase $p + 1$. The first four of these are the same as described in [12] for non-repairable system.

1. A combination of component failures does not lead to system failure in both phases p and $p + 1$.
2. A combination of component failures leads to system failure in both phases p and $p + 1$.
3. A combination of component failures does not lead to system failure in phase p but leads to system failure in phase $p + 1$.
4. A combination of component failures leads to system failure in phase p but not in phase $p + 1$.
5. Due to repair the system in a failed state may transit back to a up state.

The mechanism to compute unreliability of a system at time t , whose behavior is described using fault trees for different phases, is to compute the probabilities of all events at time t and then evaluate the fault tree using those event probabilities. The events here are whether components are up or failed. We already have described mechanism to compute the event probabilities at time t in Section 3.1. Using that we can evaluate the fault tree applicable at time t .

The first three cases listed above directly contributes towards unreliability or reliability and are taken care appropriately by a fault tree evaluation. Fault tree for a phase include failure combinations which remain common in all phases and those combinations which are considers as success earlier but are treated as failure in the current phase. Such combinations can be treated as failure combinations over all phases as the system eventually fails in phase where this combination leads to system failure. These are referred to as latent failures in [11]. Hence applying the failure criteria of the current phases to previous phases is correct and appropriate. The unreliability can be evaluated by evaluating the fault tree for current phase.

However, in order to compute correct unreliability, we must compute the probability of the system being in failed state in any phase. The fault tree evaluation for the current phase does not include the last two cases.

If a system state is a failed state up to phase p and then, it is a up state, the probability accumulated in that state up to the end of phase p must be counted towards unreliability. Such failure combinations can be identified using phase algebra as described in [12].

The only additional complication now is due to repairs as listed in case 5. We need to identify the probability that is once associated with a failed state in a previous phase but now is been associated with a success state. A straightforward evaluation of fault tree associates such probabilities with success states that get counted as reliability. We need to identify probabilities. This can be done by extending the phase algebra.

Notice that even if the success criteria remains, the last scenario must still be analyzed and accounted for. Also notice that in most cases, we assume that the components being repaired are those which are not being required for system operation in that phase. Therefore, the success criteria will not remain same over all phases.

In a Markov chain-based analysis, 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 those combinations when the system may or may not fail at the time of a phase transition.

Thus, our methodology consists of the following steps. We divide the system unreliability of a phased mission system into three parts: (i) common failure combinations; (ii) phase failure combinations, and (iii) repair to success combinations. Common failure combinations are specified by the fault tree description of the current phase. Phase failure combinations and repair to success combinations are identified using the phase algebra. These includes all those factors which describe failure in previous phases but are not considered as failure now or those flows which occurred from failed combinations to success combinations.

5.2 Phase Failure and Repair to Success Combinations

To determine phase failure and repair to success combinations for a phase p in a P phase system, we use the following procedure. Let E_p be the Boolean logic expression specifying the failure combinations for phase p . Then phase failure combinations which are treated as success combinations for all the subsequent phases and repair to success combinations for phase p , combinely denoted as (PFC_p) , are given by

$$PFC_p = (\dots((E_p \wedge \overline{E_{p+1}}) \wedge \overline{E_{i+2}}) \dots \wedge \overline{E_P}).$$

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

$$PFC_p = E_p \wedge \overline{(E_{p+1} \vee \dots \vee E_P)}.$$

The form of the expression are the same as that is given in [12]. Reader who is familiar with the work in

[12] should be careful while reading the section as there are a few differences for the algebra here from the one described in [12]. The rules for manipulating expression are different to account for repairs. In fact, they are same as applicable for Boolean algebra and the special treatment for non-repairable systems as in [12] is not required any more. Also, the computation of probability requires further attention.

5.3 Phase Algebra

Let $\bar{x} = 1$ mean that component X has failed. Then $x = 0$ implies that component X has failed and $x = 1$ means that component X is operational. Using this notation, for the system described in Figure 1, there is only one possible configuration but the component used in a phase changes from phase to phase. Thus, the following Boolean expression describe the failure for any phase. Also, the component not being used in a phase is assumed to be repaired.

$$SE(X) = \bar{x}$$

Similarly, for the system described in Figure 2 the following Boolean expressions describe the failure combinations for phases using *OR* or *AND* configurations.

$$ORE(X, Y) = \bar{x} + \bar{y}$$

$$ANDE(X, Y) = \bar{x} \bar{y}$$

Notice that X and Y are only parameters here and will be replaced by A , B , or C depending on the use of components. It should also be noted that event \bar{x} denotes the failure of component X in that phase 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 $x_p = 1$ denote the event that component X is operational during phase p . This is irrespective of the status of that component in any previous phase. With this addition, the Boolean expression for phase p for system 1 is given by the following.

$$SE_p(X) = \bar{x}_p$$

Similarly, the expressions for system 2 become

$$ORE_p(X, Y) = \bar{x}_p + \bar{y}_p$$

and

$$ANDE_p(X, Y) = \bar{x}_p \bar{y}_p$$

respectively.

Using the above two phases, it is possible that a system may be have *AND* configuration in phase p followed by *AND* or *OR* configuration in phase $p+1$ or *OR* configuration in phase p followed by *AND* or *OR* configuration in phase $p+1$. The four possible combinations *PFCs* for phase p assuming that phase $p+1$ is the last phase, components X and Y are used in phase p , and components Y and Z are used in phase $p+1$ are given in Equation 8.

$$\begin{aligned}
PFCAND(X, Y)_p OR(Y, Z)_{p+1} &= (\overline{x_p} \overline{y_p}) (\overline{y_{p+1} + z_{p+1}}) = (\overline{x_p} \overline{y_p}) (y_{p+1} z_{p+1}) \\
PFCAND(X, Y)_p AND(Y, Z)_{p+1} &= (\overline{x_p} \overline{y_p}) (\overline{y_{p+1} z_{p+1}}) = (\overline{x_p} \overline{y_p}) (y_{p+1} + z_{p+1}) \\
PFCOR(X, Y)_p OR(Y, Z)_{p+1} &= (\overline{x_p} + \overline{y_p}) (\overline{y_{p+1} + z_{p+1}}) = (\overline{x_p} + \overline{y_p}) (y_{p+1} z_{p+1}) \\
PFCOR(X, Y)_p AND(Y, Z)_{p+1} &= (\overline{x_p} + \overline{y_p}) (\overline{y_{p+1} z_{p+1}}) = (\overline{x_p} + \overline{y_p}) (y_{p+1} + z_{p+1})
\end{aligned} \tag{8}$$

When the expression for PFC_p is simplified, regular Boolean algebra rules can be applied. For this purpose, if p and q are two phases, then x_p and x_q must be treated as separate variables. The normal Boolean algebra rules such as $x_p x_p \rightarrow x_p$, $\overline{x_p} \overline{x_p} \rightarrow \overline{x_p}$, $\overline{x_p} x_p \rightarrow 0$, and their dual apply. Any product terms involving x_p or x_q or their complements must be retained as it.

An expression such as $x_p \overline{x_q}$ means that component X is operational at the end of phase p but fails by the time phase q is finished. On the other hand, an expression like $\overline{x_p} x_q$ implies that component X is failed at the end of phase p but is operational at the end of phase q due to repair carried out during the process. Thus, if $p = q - 1$ (two consecutive phases), then probability $P(x_p \overline{x_q})$ is given by $P_{Xubp} P_{Xufq}$ and probability $P(\overline{x_p} x_q)$ is given by $P_{Xfbp} P_{Xfuq}$. Other combinations are evaluated in a similar fashion. If no repair is carried out then $P_{Xfuq} = 0.0$.

5.4 System Unreliability

Using the phase success criterias for different phases and phase algebra we compute the system unreliability as follows. For a P phase system, we first compute the PFC_p 's for all phases assuming P as the last phase. Then the system unreliability is given by

$$UR = P(E_P) + \sum_{p=1}^{P-1} P(PFC_p)$$

where $P(E_P)$ is the probability of failure evaluated using the fault tree E_P of phase P (the last phase) and the failure distribution function calculated for each component as described in Section 3. $P(PFC_p)$ is the probability of phase failure combinations for phase p .

Interpretation of Boolean Expressions While computing probabilities of PFC_p 's, derived above, we may encounter expressions like $x_1 \overline{x_2} x_4 \overline{x_5}$. What it means is that we are looking for probability of a combination of events where Component X remains operational up to the end of phase 1, fails by the time phase 2 ends, but is

operational again by the end of phase 4, and then fails by the time phase 5 finishes. The following tree is useful in explaining how to compute the probability of this combination of events for component X .

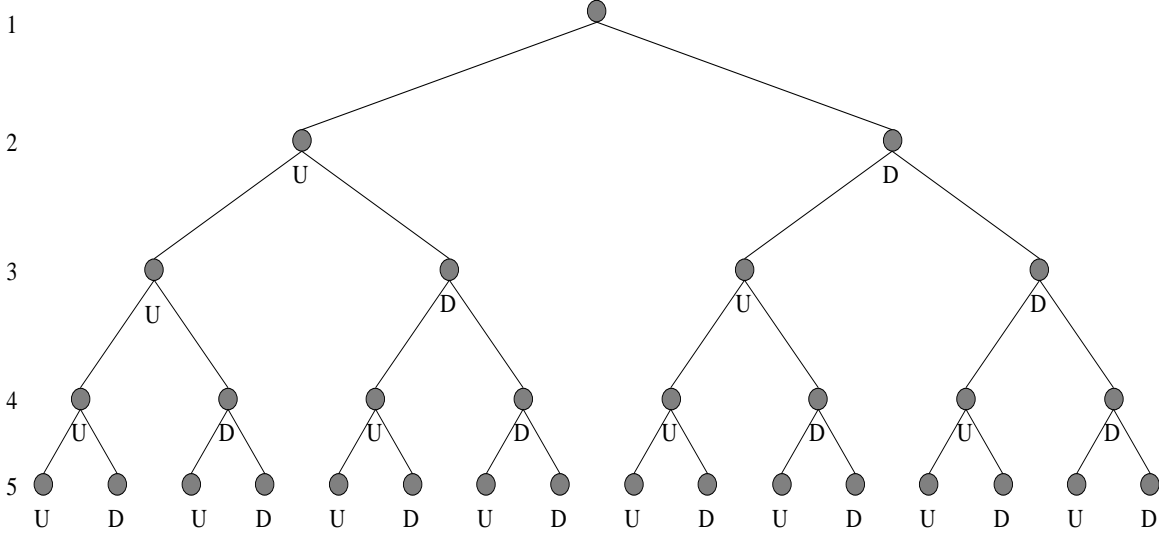


Figure 4: A component up/fail tree over multiple phases

In the tree if we assume that the root at level 1 is representing an event that component X is up at the end of phase 1 (there is certain probability associated with it), then the left child (at level 2) is representing that it is up at the end of phase 2 and the right child (at level 2) is representing that it is failed. We can compute the probabilities of these events using expressions for P_{Xu2} and P_{Xuf2} from phase 2 parameters. Similar interpretation exists for children of level 2 nodes from phase 2 to phase 3 as the component state changes. To go from Component X has failed at the end of phase 2 to the state that it is operational at the end of phase 4, there are two routes, i.e., $\bar{x}_2 \rightarrow \bar{x}_3 \rightarrow x_4$ and $\bar{x}_2 \rightarrow x_3 \rightarrow x_4$. We need to compute the probabilities of both paths and then add them up to arrive at the probability of combination \bar{x}_2x_4 .

We may encounter any combination of such events for a component but it should be obvious that such computations are required to be done for each component and not for system states. For a component, if there are p phases, then there at most 2^{p+1} values which we need to store. In an N component system, this amounts to $N2^{p+1}$ values. On the other hand in a system with N components, there could be up to 2^N states and we have to analyze them for p phases. So we may be storing up to $p2^N$ states combination. Normally, $N \gg p$ (will not be the case for examples in the paper for the obvious reasons). Thus the technique here is computationally much more efficient than generating a state space and computing state occupation probabilities for those states for each phase given a distribution from a previous phase operation.

5.5 Computing Transient Behavior

In the previous section, we outlined the mechanism to compute unreliability at the end of a mission, that is, the end of the last phase. Sometime one may be interested in computing the unreliability behavior during all phases. This means we need to compute unreliability for each phase as a function of time. It turns out that this is not expensive and can be easily accommodated in our methodology as the *PFCs* calculation is recursive.

Recall that *PFCs* for a phase are computed as

$$PFC_p = E_p \wedge \overline{(E_{p+1} \vee \dots \vee E_P)}.$$

Also, the unreliability at the end of a mission is computed using the expression

$$UR = P(E_P) + \sum_{p=1}^{P-1} P(PFC_p).$$

In a P phase system, we define $PFC_P = E_P$ then the unreliability for a P phase system can be written as

$$UR = \sum_{p=1}^P P(PFC_p).$$

Thus, to compute unreliability at the end of phase p , we need $PFC_1, PFC_2, \dots, PFC_p$ where the *PFCs* must be calculated using phase p as the last phase. We define $PFC_{i,p}$ as the *PFC* of phase i , $i < p$, assuming phase p as the last phase. Then the following relation holds.

$$PFC_{i,p} = PFC_{i,p-1} \wedge \overline{E_p}$$

The unreliability of the p th phase is computed by using the following relation.

$$UR_p = \sum_{i=1}^p P(PFC_{i,p})$$

and the $PFC_{i,p}$ can be computed recursively using the results of $PFC_{i,p-1}$ and E_p . With this recursive relation, one may compute reliability of phase p using the result of phase $p - 1$.

5.6 Latent Failures

It should also be noticed that at the transition of a phase, one may see a upwards change in unreliability value at the phase transition time. This happens if the next phase has different success criteria than the current phase. In that case it is possible that that some of the success states in phase i may be failed states in phase $i + 1$. We define them as latent failures as the system may fail as soon as the phase change occurs. For example, in an automobile system, on a freeway we may be cruising at a fixed speed and we may not need the brake subsystem

in a car. But as soon as we hit a city limit, a phase change occurs and if the brakes are not fully functional, we are likely to hit some other vehicle. To compute unreliability increase due to phase change from phase i to phase $i + 1$, we compute UR_i . Then, we compute UR_{i+} which is just after the end of phase i and beginning of phase $i + 1$. For this purpose, we modify the success criteria and it is now a logical sum of the success criterias of phases i and $i + 1$ evaluated at the end of phase i using parameters of phase i . We define this as $L_i = E_i + E_{i+1}$ with E_{i+1} specified using component status at the end of phase i . PFCs also need to be reevaluated as L_i instead of E_i for the phase i (for earlier phases, we will still use E_p and not L_p for $p < i$).

We will demonstrate our methodology using the examples described above in the following section.

5.7 Example Computations

In the first example, we use the two component system with four phases. In the first phase, we require component A for operation (and therefore there is no repair on it, see discussion above in Section 4). Component B has associated with it both failure and repair rates. Then we alternate between the use of component and repair. Thus the success criterias for four phases are specified by

$$E_1 = SE_1(A) = \overline{a_1}; \quad E_2 = SE_2(B) = \overline{b_2}; \quad E_3 = SE_3(A) = \overline{a_3}; \quad E_4 = SE_4(B) = \overline{b_4}. \quad (9)$$

Using the above information, at the phase changes from p to $p + 1$, there could be latent failure (they are in this system) and to evaluate unreliability including phase change boundary, we will use L_i instead of E_i as discussed above. The success criteria with latent failures is given by

$$L_1 = SE_1(A) + SE_1(B) = \overline{a_1} + \overline{b_1}; \quad L_2 = SE_2(B) + SE_2(A) = \overline{b_2} + \overline{a_2}; \quad L_3 = SE_3(A) + SE_3(B) = \overline{a_3} + \overline{b_3}. \quad (10)$$

We assume that there is no phase change after phase 4. Using this information we can compute PFCs as follows.

$$\begin{aligned} PFC_{12} &= (E_1 \cdot \overline{E_2}) &&= \overline{a_1}b_2 \\ PFC_{13} &= (PFC_{12} \cdot \overline{E_3}) &&= \overline{a_1}b_2a_3 \\ PFC_{23} &= (E_2 \cdot \overline{E_3}) &&= \overline{b_2}a_3 \\ PFC_{14} &= (PFC_{13} \cdot \overline{E_4}) &&= \overline{a_1}b_2a_3b_4 \\ PFC_{24} &= (PFC_{23} \cdot \overline{E_4}) &&= \overline{b_2}a_3b_4 \\ PFC_{34} &= (E_3 \cdot \overline{E_4}) &&= \overline{a_3}b_4 \end{aligned} \quad (11)$$

Now to compute, latent PFCs (that is including latent failures at the phase transition points), we use the same expressions except that we need to L_i instead of E_i and obtained the following LPFCs. Notice that in the recursive function, we continue to use PFC and L_i is only used for the current last phase.

Table 1: State Probabilities and Unreliabilities for a two component system

State	BP1	EP1	BP2	EP2	BP3	EP3	BP4	EP4
Factor	1.000	1.000	1.000	0.891	0.891	0.891 ²	0.891 ²	0.891 ³
11	1.000	0.891	0.891	0.891	0.891	0.891	0.891	0.891
10	0.000	0.009	0.000	0.099	0.000	0.009	0.000	0.099
10	0.000	0.099	0.000	0.009	0.000	0.099	0.000	0.009
00	0.000	0.001	0.000	0.001	0.000	0.001	0.000	0.001
UR	0.000	0.100	0.109	.1981	.206119	0.2855071	0.29265203	0.36338683

$$\begin{aligned}
 LPFC_{12} &= (E_1 \cdot \overline{L_2}) &= \overline{a_1} a_2 b_2 \\
 LPFC_{13} &= (PFC_{12} \cdot \overline{L_3}) &= \overline{a_1} b_2 a_3 b_3 \\
 LPFC_{23} &= (E_2 \cdot \overline{L_3}) &= \overline{b_2} a_3 b_3
 \end{aligned} \tag{12}$$

Then the unreliability at the end of phase p and at the beginning of phase $p + 1$ is given by the following expressions.

$$\begin{aligned}
 UR_p &= \sum_{i=1}^{p-1} P(PFC_{i,p}) + P(E_p) \\
 LUR_p &= \sum_{i=1}^{p-1} P(LPFC_{i,p}) + P(L_p)
 \end{aligned} \tag{13}$$

We computed numerical results using above expressions and parameters values which are easy to verify by hand computation. We first used phase durations for each phase as 10 hours and value of failure and repair rates for both components in such a way that the factor α at phase duration of 10 hours is equal to 0.9. Also, if repair is applicable, then parameter β in all phases for applicable components is also 0.9. Using, these parameter values, we get the results shown in Table 1. Here BP and EP stands for beginning of phase and end of phase and we are tabulating SOP for each state, reliability, and unreliability and we have a multiplication factor associated with all column entries. Idea is to be able to clearly see that the results are correct. The results are obtained using SHARPE [2] program where PFC expressions were hand coded, EHARP [10], and hand calculations. the results match in all cases to 9 significant digits. The multiplication factor only applies to SOPs and the unreliability values are as they are listed.

To give a better idea appreciation for results and match the results of this table to that obtained using Markov chain analysis, the Markov chains and the initial state occupation probabilities for four phases are shown in Figure 5. Any state occupation probability not shown is zero (that is the case for three states out of four in every phase). Two of the states are failure states in each phase. One of the remaining two states becomes a latent failure state. Thus only one state is operational state at the beginning of each phase.

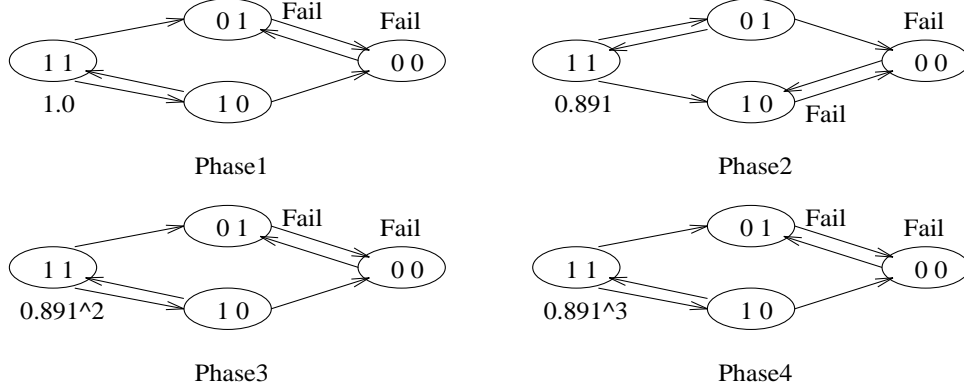


Figure 5: Markov Chains for four phases with initial SOPs

Table 2: Unreliabilities for a two component system (variable parameters)

Case	EP1	BP2	EP2	BP3	EP3	BP4	EP4
1 ($\times 10^{-4}$)	0.99995000	1.63198093	2.63176774	3.26369553	4.26331917	4.89514383	5.89460434
2 ($\times 10^{-4}$)	0.99995000	1.99980001	2.99955004	3.99920011	4.99875021	5.99820036	6.99755057
3 ($\times 10^{-3}$)	0.99950016	1.09938570	2.09778703	2.19756275	3.19486645	3.29453247	4.29073975
4 ($\times 10^{-3}$)	0.99950016	1.99800133	2.99550450	3.99201066	4.98752081	5.98203595	6.97555707
5 ($\times 10^{-4}$)	0.99995000	1.06315547	2.06299916	2.12619791	3.12593531	3.18912734	4.18875844
6 ($\times 10^{-4}$)	0.99995000	1.09993950	2.09977952	2.19975802	3.19948805	3.29945556	4.29907563
7 ($\times 10^{-3}$)	0.99950016	1.00948962	2.00798080	2.01796017	3.01544338	3.02541268	4.02188894
8 ($\times 10^{-3}$)	0.99950016	1.09939522	2.09779654	2.19758177	3.19488546	3.29456098	4.29076824

Next we used other data to compute the results. In all cases the repair rate if applicable remains to be 0.100/hour. In the first four cases, we use failure rate of each component irrespective of usage as 0.00001/hour. In the last four cases, we use failure rates of used components as 0.00001/hour while those under repair as 0.000001/hour. The phase durations for cases 1, 2, 5, and 6 are 10 hours while in other four cases, 3, 4, 7, and 8, are 100 hours. In even number cases, the analysis is done by ignoring repairs while odd cases include repairs. Table 2 contain the results obtained in all cases.

First notice the multiplication factors for each row. A factor of 10 difference is there due to the mission (phase) times. Next, when we ignore repairs, we notice a substantial change in unreliability values obtained in the first four cases when the failure rates are the same whether a component is being repaired or not. Thus repairs must be accounted for in such cases. More interesting results are obtained when the components being repaired have an order of magnitude smaller failure rates (cases 5-8). In these cases, ignoring repairs impacts the

results but in this example the difference is not substantial. So one may choose one vs another analysis based on parameter values.

Example 2. For example 2, we consider the three components, A , B , and C , system with two phase configurations AND and OR and three phases. In each phase one component is not used. Suppose component A is not used in phase 1, component B is not used in phase 2, and component C is not used in phase 3. There are eight possible combinations (AND or OR in each phase). We will not write expressions for $PFCs$ and $LPFCs$ for all cases here. But to demonstrate how to derive them, for one case when Phase 1 is $OR(B, C)$, phase 2 is $AND(C, A)$ and phase 3 is $AND(A, B)$. Then

$$PFC_{12} = PFCOR(B, C)_1AND(C, A)_2 = (\bar{b}_1 + \bar{c}_1)(c_2 + a_2)$$

and

$$PFC_{23} = PFCAND(C, A)_2AND(A, B)_3 = (\bar{c}_2 \bar{a}_2)(a_3 + b_3)$$

as computed in Equation 8. We can also compute PFC_{13} using the recurrence relation to obtain

$$PFC_{13} = PFC_{12}\bar{E}_3 = (\bar{b}_1 + \bar{c}_1)(c_2 + a_2)(a_3 + b_3).$$

To compute the probabilities of these expressions, we need to expand the expression in mutually exclusive terms. It should be noted that when expressions are in product of expressions form, each product expression can be independently expanded into mutually exclusive terms. Then a product expansion will give all terms which are mutually exclusive. So using this, we compute probabilities of $PFCs$ as given below for this case.

$$\begin{aligned} P(PFC_{12}) &= P((\bar{b}_1 + \bar{c}_1)(c_2 + a_2)) = P((\bar{b}_1 + b_1\bar{c}_1)(a_2 + \bar{a}_2c_2)) \\ &= P(a_2\bar{b}_1) + P(\bar{a}_2\bar{b}_1c_2) + P(a_2b_1\bar{c}_1) + P(a_2b_1\bar{c}_1c_2) \\ P(PFC_{13}) &= P((\bar{b}_1 + \bar{c}_1)(c_2 + a_2)(a_3 + b_3)) = (\bar{b}_1 + b_1\bar{c}_1)(a_2 + \bar{a}_2c_2)(a_3 + \bar{a}_3b_3) \\ &= P(a_2a_3\bar{b}_1) + P(a_2a_3b_1\bar{c}_1) + P(a_2\bar{a}_3\bar{b}_1b_3) + P(a_2\bar{a}_3b_1b_3\bar{c}_1) \\ &= +P(\bar{a}_2a_3\bar{b}_1c_2) + P(\bar{a}_2a_3b_1\bar{c}_1c_2) + P(\bar{a}_2\bar{a}_3\bar{b}_1b_3c_2) + P(\bar{a}_2\bar{a}_3b_1b_3\bar{c}_1c_2) \\ PFC_{23} &= P((\bar{c}_2\bar{a}_2)(a_3 + b_3)) = \bar{a}_2\bar{c}_2(a_3 + \bar{a}_3b_3) \\ &= P(\bar{a}_2\bar{c}_2a_3) + P(\bar{a}_2\bar{c}_2\bar{a}_3b_3) \end{aligned} \tag{14}$$

We programmed each of the eight possible cases. We used failure rate for each component to be 0.0001/hour and repair rate to be 0.1/hour wherever applicable in a 10 hours/phase mission. The results for eight cases are shown in Table 3. Here in phase name ‘‘A’’ means AND phase and ‘‘O’’ means OR phase. Then, we assumed that the failure rate for the component under repair is small, i.e., 0.00001/hour and recomputed all the eight cases. These results are in Table 4. One can notice the difference in unreliability in the two cases. We are not showing the results when we ignore the repairs altogether but, we noticed that the difference is significant in the first case and relatively less in the second case.

Table 3: Unreliability for eight cases with same failure rates

Case	EP1	BP2	EP2	BP3	EP3
AAA	9.99000583e-07	1.62990993e-06	4.25556226e-06	5.88170181e-06	9.49979360e-06
OAA	1.99800133e-03	1.99800133e-03	1.99962799e-03	2.00065528e-03	2.00390329e-03
AOA	9.99000583e-07	1.63072502e-03	3.62546817e-03	3.62546817e-03	3.62745761e-03
OOA	1.99800133e-03	2.62859528e-03	4.62134468e-03	4.62134468e-03	4.62296705e-03
AAO	9.99000583e-07	1.62990993e-06	4.25556226e-06	2.62891027e-03	4.62165904e-03
OAO	1.99800133e-03	1.99800133e-03	1.99962799e-03	4.62239334e-03	6.24453356e-03
AOO	9.99000583e-07	1.63072502e-03	3.62546817e-03	4.62103010e-03	6.60979861e-03
OOO	1.99800133e-03	2.62859528e-03	4.62134468e-03	5.25028105e-03	7.23779231e-03

Table 4: Unreliability for eight cases with low failure rates for components while under repair

Case	EP1	BP2	EP2	BP3	EP3
AAA	9.99000583e-07	1.06211526e-06	3.12110793e-06	3.57805367e-06	6.06492674e-06
OAA	1.99800133e-03	1.99800133e-03	1.99906133e-03	1.99912829e-03	2.00124603e-03
AOA	9.99000583e-07	1.06264640e-03	3.05852457e-03	3.05852457e-03	3.05994942e-03
OOA	1.99800133e-03	2.06108445e-03	4.05496774e-03	4.05496774e-03	4.05602555e-03
AAO	9.99000583e-07	1.06211526e-06	3.12110793e-06	1.49368754e-03	3.48870448e-03
OAO	1.99800133e-03	1.99800133e-03	1.99906133e-03	3.48887187e-03	5.11330514e-03
AOO	9.99000583e-07	1.06264640e-03	3.05852457e-03	3.48807495e-03	5.47910711e-03
OOO	1.99800133e-03	2.06108445e-03	4.05496774e-03	4.11792084e-03	6.10769456e-03

Table 5: Unreliability for “all is well if end is well” case

Case	EP1	BP2	EP2	BP3	EP3
$\alpha\beta\gamma R$	1.89437172e-03	1.89437172e-03	2.52542938e-03	2.52542938e-03	3.38726223e-03
$\alpha\beta\gamma N$	2.99550450e-03	2.99550450e-03	3.99300567e-03	3.99300567e-03	5.97905190e-03
$\gamma\beta\alpha R$	2.52263933e-10	6.32255388e-04	8.64817157e-04	2.58997399e-03	3.39046756e-03
$\gamma\beta\alpha N$	9.98501249e-10	1.00049817e-03	2.00198537e-03	5.98203595e-03	8.95962123e-03

Example 3. In our last example, we programmed the third case where the three phases are $\alpha = OR$, $\beta = OR - AND$, and $\gamma = OR$ as shown in Figure 3. We ran four cases for this example. These had two orders $\alpha\beta\gamma$ and $\gamma\beta\alpha$ and in each case there is repair on all components in all phases (R) or no repair on any component (N). The phases are each of 10 hours durations. The failure rates for each component in each phase is 0.0001/hour. The repair rates for each component when applicable is 0.1/hour. The results are shown in Table 5. Notice two things. Once ignoring repairs have significant impact on unreliability due to repairs, in particular for the system where the success criteria is more stringent during the later phases. With repairs, the unreliability can be almost maintained at the same levels as is the case in the first and the third line.

6 Managing Phased-Mission Systems with Repairs Using RBDs

It should be mentioned that this analysis can also be carried out using RBDs. Recall that in [5] each component X model in phase p is replaced by a series of events $x_1x_2 \cdots x_p$. In case of repairs, each component model will be a parallel series model derived out of component up/fail tree as shown in Figure 4. There will be up to 2^{p-1} parallel branches. Each branch represents one unique path from root to one of the leaf U node in the tree. Notice that if a particular phase does not have repair on a particular component, then the tree does not have any expansion from that the intermediate D node in the tree. The rest of the analysis remains the same.

7 Conclusions

We have presented a technique to analyze phased-mission systems including component repairs whose phase success criterias can be expressed using fault trees. This technique yields accurate results and is simple in concept and computation. For this purpose, we enhanced phase algebra to include the effects of phases that allows us to efficiently compute the probabilities of all possible combinations contributing to failure in phased-mission systems during individual phases. This technique is very useful for a large class of systems where during the long mission times the system includes repairs but system operational behavior can be described using fault trees. Several examples have been included to show the effects of repairs and how to manage it computationally. Currently we are incorporating these techniques in reliability analysis tools.

References

- [1] K. Trivedi, J. Dugan, R. Geist, M. Smotherman, B. Rothmann, M. Boyd, and S. Bavuso, "HARP: Introduction and guide for the users," Dept. of Computer Science, Duke University, Durham, NC 27706, March 1986.
- [2] R. Sahner and K. Trivedi, "Reliability Modeling using SHARPE," *IEEE Trans. Reliability*, vol. R-36, June 1987, pp. 186-193.
- [3] A. Goyal, W. C. Carter, E. de Douza e Silva, and S. S. Lavenberg, "The System Availability Estimator," in *Proc. of FTCS-16*, June 1986, pp. 84-89.
- [4] R. Butler, "The SURE Reliability Analysis Program," *AIAA Guidance, Navigation, and Control Conference*, Williamsburg, Virginia, August 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.
- [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. R-40, 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. K. Somani, J. A. Ritcey, and S. Au, "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. K. Somani, S. Palnitkar, and T. C. Sharma, "Reliability Modeling of Systems with Latent failures using Markov Chains," *Proc. of RAMS-93*, pp. 120-125.
- [12] A. K. Somani and K. S. Trivedi, "Phased-Mission System Analysis Using Boolean Algebraic Methods," in the *Proc. of Sigmetrics*, 1994, pp. 98-107.
- [13] 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.

- [14] K. S. Trivedi, "Probability & Statistics with Reliability, Queuing, and Computer Science Applications," Prentice-Hall, INC, Englewood Cliffs, NJ.