NASA Contractor Report 159358

COMPARATIVE ANALYSIS OF TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS

E. F. Hitt, M. S. Bridgman, and A. C. Robinson

BATTELLE'S COLUMBUS LABORATORIES
505 King Avenue
Columbus, Ohio 43201

CONTRACT NAS1-15760
APRIL 1981

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

FINAL REPORT

on

COMPARATIVE ANALYSIS OF TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS

May 12, 1981

CONTRACT NO. NAS1-15760

BATTELLE
Columbus Laboratories
505 King Avenue
Columbus, Ohio 43201

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

TABLE OF CONTENTS (Continued)

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF FIGURES (Continued)

COMPARATIVE ANALYSIS OF TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS

Ellis F. Hitt, Michael S. Bridgman, and Alfred C. Robinson

BATTELLE
Columbus Laboratories

SUMMARY

The objective of this study was to evaluate "performability", a technique developed by the University of Michigan under NASA Grant NSG 1306, for its accuracy, practical usefulness, and cost of use. Performability analysis determines the probabilities of occurrence for a set of mission outcomes. It was designed for application to fault-tolerant computing systems used in multiphase missions. Performability was found to require significantly more time to learn and understand that the fault-tree method.

Performability and the fault trees were applied to a set of sample problems ranging from simple to moderately complex in nature. The problems involved up to five outcomes, two to five mission phases, permanent faults, and some functional dependencies. Two to six times as much clock time was required to apply performability as fault trees. Much of the performability effort was mechanical in nature. More ingenuity was required for the fault-tree solutions. Initial results from the methods often disagreed. Detailed analyses revealed the results were sensitive to mathematical procedures followed in dealing with small differences, round-off procedures, programming procedures, and the computer used. The use of only one method would not have revealed this sensitivity. As an observation, both methods appear to provide more precision than can be supported by available data.

For most problems of practical interest, fault trees will be more useful than performability analysis. For highly complex problems, performability may offer advantages in solution accuracy and required solution effort. If performability analysis is to be further developed, then tutorial material should be written, the probability computation program should be validated, and further mechanization of the technique should be investigated.

LIST OF STANDARD SYMBOLS

$\{ \quad \}$	Set
$\cup$	Union
$\cap$	Intersection
$\sum$	Summation
$\prod$	Product
P( ) or Pr( )	Probability
E( )	Expected value
γ^{-1}	Inverse of the function γ
ε	Element of (a set)

INTRODUCTION

Various techniques exist for evaluating the effectiveness of aircraft computing systems. These techniques have been used for assessing primarily the reliability and safety of flight control systems and digital avionics. The techniques are generally mathematical models which may be manually applied or may be implemented in computer programs. These models are normally used rather than testing techniques to determine the reliability to avoid the cost of performing reliability testing.

With the development of fault tolerant computing systems, testing becomes even more impractical because of the many fault tolerant architectural concepts that are possible and the fact that testing requires that the system design be committed to hardware and software. Techniques are required that can be used to design fault tolerant computing systems as well as evaluate the design of candidate fault tolerant computer systems prior to the actual development of the hardware and software which implement the candidate design. New techniques such as that developed by the University of Michigan under NASA Grant NSG-1306 must be evaluated against a proven technique prior to widespread application in order to assure that the results obtained are valid. This in itself poses a problem, since many of the proven techniques either are unwieldy and very time consuming when applied to systems of moderate complexity, or do not properly treat software errors, transient failures, and other features of fault tolerant systems. The total system must be analyzed and not just a portion such as the hardware components or the software. The nature of the systems to be analyzed are categorized by the complexity of relationships among system elements under the control of a software executive program. This complexity can lead to an intractable analysis problem for a completely general system. Many of the techniques, such as that developed under NASA Grant NSG-1306, assume some simplification by combining or partitioning system states.

The objective of this report is to present the results of an evaluation of the practical usefulness of the techniques developed under NASA Grant NSG-1306 compared to other techniques such as the "conventional" fault tree analysis. These comparative analyses were made based upon data obtained from actual application of the techniques to hypothetical systems in realistic

mission environments. The sample problems were solved using the NASA Grant NSG-1306 techniques (referred to hereafter in this report as "performability analysis"), fault tree analysis, and the Tabular System Reliability Analysis (TASRA).

The first-level problem is a simple series-parallel problem which was used primarily to verify the researcher's understanding of the respective techniques and to obtain a preliminary comparison of the relative ease with which the techniques could be applied to a simple problem not involving time or environmental dependency. The results are basically a reliability measure involving both levels of component failures and degraded component performance.

The second problem considered by the analysts involved a dual-dual flight control system and a simple mission scenario consisting of a takeoff/climb phase, a cruise phase, and a descent and landing phase with Category II weather at the scheduled destination.

The third problem required the analysts to analyze a digital flight control system which possessed some of the features of the Fault Tolerant Multi-Processor (FTMP) architecture developed by C. S. Draper Laboratories.

The objective of all analyses was to provide a comparison of the techniques for each of the three problems. This comparison involves assessment of the comparative and absolute difficulty in applying the techniques to arrive at the cost measure including the staff time and costs involved in learning the techniques as well as the staff time and costs involved in applying the techniques to each of the problems.

This report presents a synopsis of the techniques considered, a description of the fault tolerant system designs analyzed and the scenarios for each of the problems, and a comparative analysis of the results obtained by each analyst for each of the system designs analyzed. The final section of the report presents the conclusions and recommendations based upon the analyses performed.

SYNOPSIS OF TECHNIQUES

SYNOPSIS OF PERFORMABILITY ANALYSIS

Performability analysis is the name given to a technique for evaluating the effectiveness of aircraft computing systems. The technique has been under development at the University of Michigan since November 1976 as a research project for NASA Langley Research Center under NASA Grant NSG-1306. This brief synopsis of performability analysis is intended to summarize the technique and to establish pertinent definitions. No attempt is made to explain the theoretical development or to explore the more sophisticated aspects and capabilities of the technique. Detailed material on performability analysis is contained in References 1-9.

Overview

Consider an aircraft computing system used in a multiphase mission. The system user (e.g., the airline) can define a set of mission outcomes, which is called the "accomplishment set". The accomplishment set has the form $A = \{a_o, a_1, .., a_n\}$ where the a_i are "accomplishment levels". An example accomplishment level is "safe, on-time, fuel-efficient flight". The "performability" of the system is the set of probabilities of realizing each of the accomplishment levels. In mathematical terms, the performability is

$$P(a_o),\ P(a_1),\ \ldots,\ P(a_n)$$

where $P(a_i)$ = probability of outcome a_i occurring.

On a detailed (i.e., component) level, the system behavior is viewed as a stochastic process : $X_s = \{X(t) \mid t \varepsilon T\}$ where X_t is the state of the system (e.g., a computer and its environment) at time t and T is the set of times at which the system is observed. For a mission with m phases, observations can be made at time zero (t_o) and at the end of each phase (t_j, $j = 1,2, \ldots, m$). Let Q represent the state space of the system. Then each X(t) is an element of Q. Let $q_j = X(t_j)$. A particular instance of system behavior is given by the "trajectory"

$$\mu = (q_o, q_1, q_2, \ldots, q_m).$$

The space of all possible trajectories is called the "trajectory space" and is denoted by U.

Each trajectory $\mu \varepsilon U$ corresponds to a single mission outcome $a \varepsilon A$. This mapping is denoted as follows:

$$\gamma : U \rightarrow A.$$

and γ is called the "capability function".

The two basic steps of performability analysis are:

Step 1. For each accomplishment level $a \varepsilon A$, find
$\gamma^{-1}(a)$ = set of all trajectories $\mu \varepsilon U$ which result in the outcome "a".

Step 2. For each $a \varepsilon A$, compute the probability of occurrence of $\gamma^{-1}(a)$. Then,

$$P(a) = Pr[\gamma^{-1}(a)].$$

These steps are explained in more detail in the following subsections.

Step 1. Find $\gamma^{-1}(a)$

For simple systems, the set $\gamma^{-1}(a)$ can be determined by inspection of the base model trajectories. As more complexity (e.g., more components, phases, interdependencies, outcomes) is introduced, it becomes increasingly difficult to determine $\gamma^{-1}(a)$ in a single step. A hierarchy of models can be used to determine the capability function (i.e., to connect the base model trajectory space U with the accomplishment set A).

While any number of intermediate models could be used, this discussion uses two: a mission model, also called the "level 0" model, and a function, or "level 1", model. The base model is called the component or "level 2" model. Figure 1 summarizes the model hierarchy. Each level has an associated trajectory space which describes the possible mission profiles in terms of the state space for that level. The level 0 state space could consist of parameters representing such mission characteristics as safety, economics, and/or operations. For level 1, the state space could consist of the functions to be accomplished, such as flight augmentation, navigation, and flight control. It could also include environmental variables such as the weather at the destination. The base level model could then be expressed in terms of the components which comprise the system. Model level j is related to the next "lowest" level (j-1) by a function denoted K_j. These functions are defined by the nature of the system and its mission requirements.

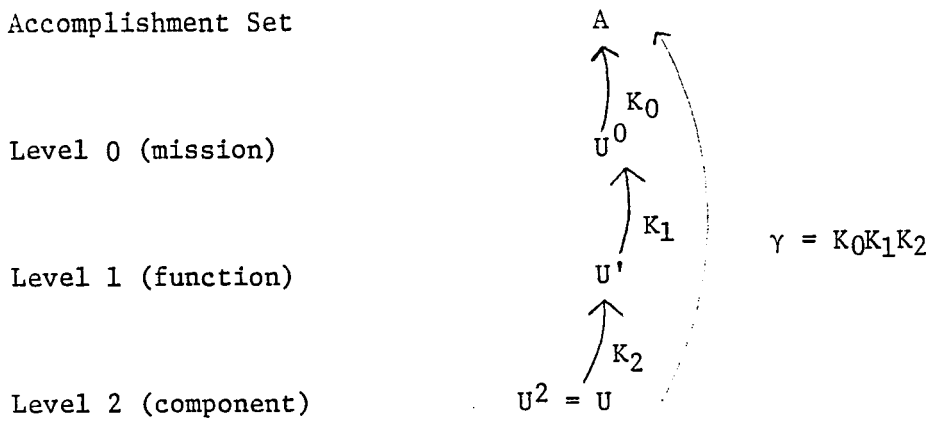


FIGURE 1. SUMMARY OF PERFORMABILITY ANALYSIS MODEL HIERARCHY

The set γ^{-1} (a) is formulated by sequentially constructing the inverses K_j^{-1} and linking them together. For each base model trajectory $\mu \varepsilon U^2$,

$$\gamma(\mu) = (K_0 K_1 K_2)\ (\mu) \ \varepsilon\ A$$

and

$$\gamma^{-1}\ (a) = (K_0 K_1 K_2)^{-1}\ (a) \ \varepsilon\ U \quad .$$

For each mission outcome $a \varepsilon A$, first find

$$K_0^{-1}\ (a) = \text{set of all mission level trajectories in } U^0 \text{ which result in outcome "a"}$$
$$= \{w \varepsilon U^0 \mid K_0\ (w) = a\},$$

Next, for each $w \varepsilon K_0^{-1}$ (a), find

$$K_1^{-1}\ (w) = \text{set of all function level trajectories in } U^1 \text{ which result in mission trajectory w.}$$

Taking the union of these sets for all $w \varepsilon K_0^{-1}$ (a) gives

$$K_1^{-1}\ (K_0^{-1}\ (a)) = (K_0 K_1)^{-1}\ (a)$$
$$= \text{set of all function level trajectories in } U^1 \text{ which result in outcome "a".}$$

In a similar fashion, find

$$K_2^{-1}\ ((K_0 K_1)^{-1}\ (a)) = (K_0 K_1 K_2)^{-1}\ (a)$$
$$= \text{set of all base level trajectories in } U^2 \text{ which result in outcome "a".}$$
$$= \gamma^{-1}\ (a).$$

Each single-step inverse is accomplished using "projection" functions. The determination of $(K_0 K_1)^{-1}$ (a) given K_0^{-1} (a) will be used as an example. All trajectories are expressed as matrices. (Vectors and single variables are special cases of matrices.) Let $w \varepsilon K_0^{-1}$ (a) and let c_ℓ denote the ℓ^{th} component of w. The ℓ^{th} projection function, denoted ξ_ℓ, simply maps the matrix w onto its ℓ^{th} component, c_ℓ. The first need is to determine, for each component of w, the set

$$(\xi_\ell K_1)^{-1}\ (c_\ell) = \{\text{all trajectories } U^1 \text{ which, when mapped to } U^0, \text{ have the value } c_\ell \text{ for the } \ell^{th} \text{ component}\}$$
$$= \{\nu \ \varepsilon\ U^1 \mid \xi_\ell\ (K^1(\nu)) = c_\ell\}.$$

The intersection of these sets for all components of $w \varepsilon K^0$ is the set of all trajectories in U^1 which, when mapped to U^0, have c_1 for the first component, c_2 for the second component, and so on. This is exactly the set

of all trajectories in U^1 which map to $w \in U^0$. Symbolically,

$$K_1^{-1}(w) = \bigcap_{\text{All } \ell} (\xi_\ell K_1)^{-1}(c_\ell).$$

Computing the $(\xi_\ell K_1)^{-1}(c_\ell)$ sets requires knowledge of the system, its environment, and the mission. Computing the intersection is a purely mechanical process. The inverse image $K_1^{-1}(w)$ can thus be found for every $w \in K_0^{-1}(a)$. The union of all these inverse images is $(K_0K_1)^{-1}(a)$.

Step 2. Compute Pr $[\gamma^{-1}(a)]$

The first step was to determine $\gamma^{-1}(a)$, the set of base model trajectories which result in the mission outcome "a", for every $a \in A$. In this step, the probability of the set $\gamma^{-1}(a)$ occurring is computed. The method for performing the computations uses the fact that each inverse image of an element (e.g., $K_1^{-1}(w)$) is a Cartesian set*. Furthermore, each inverse of the outcome "a" (e.g., $(K_0K_1)^{-1}(a)$) is a union of disjoint Cartesian sets. The inverse image of "a" in the base model trajectories can be written as

$$\gamma^{-1}(a) = V_1 \cup V_2 \cup \ldots \cup V_s$$

where each V_i is Cartesian and $V_i \cap V_j = \phi$ for $i \neq j$.
Hence, Pr $[\gamma^{-1}(a)]$ = Pr (V_1) + Pr (V_2) + ... + Pr (V_s)
and each Pr (V_i) must be computed.

Suppose V is a Cartesian set and there are m phases in the mission. Then V can be written as

$$V = R_1 \times R_2 \times \ldots \times R_m$$

where each R_k is a subset of the state space of the system. Assume there are n possible states in the state space. The initial state vector is

$$I(o) = [p_o(1)\ p_o(2)\ \ldots\ p_o(n)]$$

where $p_o(i)$ = probability the system is in state i at the start of the mission. The intraphase transition matrix for phase k gives the state transition probabilities for the state space:

*Definition of a Cartesian set: Let Q be some set and let V be a subset of QxQ; that is, every element of V is of the form (q_1, q_2) where $q_i \in Q$. If there exist two subsets of Q, say R_1 and R_2, such that every element of V is of the form (r_1, r_2) where $r_1 \in R_1$ and $r_2 \in R_2$, and every combination (r_1, r_2) is in V, then V is Cartesian.

$$P_k = [\, p_k\ (i_1 j)]$$

where p_k (i, j) = probability the system is in state j at the end of phase k given the system was in state i at the start of phase k. If the base model is a Markov process, then the p_k (i, j) are the Markov transition probabilities.

The "characteristic matrix" for the set V and phase k is

$$G_{V,k} = [g_{v,k}\ (i,j)\ ]$$

where

$$g_{v,k}\ (i,\ j) = \begin{cases} 1 \text{ if } i = j \text{ and if state } i \text{ is in set } R_k \\ 0 \text{ otherwise.} \end{cases}$$

Multiplying the intraphase transition matrix P_k on the right by $G_{V,k}$ puts zeroes in those columns of P_k which correspond to states not in R_k (and therefore not in the set of trajectories which comprise V). In other words, $G_{V,k}$ selects those columns of P_k corresponding to the phase k outcomes in the trajectory subspace V.

For the last (m^{th}) phase, the characteristic matrix becomes the vector

$$F(m) = \begin{bmatrix} f(1) \\ f(2) \\ \cdot \\ \cdot \\ \cdot \\ f(m) \end{bmatrix}$$

where

$$f(i) = \begin{cases} 1 \text{ if state } i \text{ is in } R_m \\ 0 \text{ otherwise.} \end{cases}$$

The use of a column vector is to sum the probabilities of being in any of the acceptable final states.

Using these quantities, the probability of V is:

$$Pr(V) = I(o)(P_1 G_{1,v})(P_2 G_{2,v}) \ldots (P_{m-1} G_{m-1,v})(P_m F(m)).$$

Summary

This synopsis of performability analysis summarizes the nature of the technique. It does not address all of the capabilities or important aspects of the technique, some of which are:

- o Transitions between phases
- o "Lumping" of states to reduce the number of states needed for the computations
- o Modeling of non-Markov stochastic processes.

For further details, the reader is directed to References 1, 2, and 3.

THE FAULT TREE METHOD

Fault trees have been widely used in many types of reliability analysis, since development of the technique in the early 1960's. The major area of application has been the study of safety problems in nuclear reactors[10,11]*. A general review of applications and computational aids is given by Fussell, Powers, and Bennetts[12]. The technique is conceptually quite simple, though application to realistic problems may be laborious.

There are two aspects of the methodology which will be discussed separately: (1) construction of the fault itself and (2) computation of the probabilities of the events considered. In some applications, only the first aspect is used. In the present study, both are required.

Construction of the Fault Tree

The starting point for each fault tree is the selection of some particular event (usually an undesirable event) for study. In most problems there is more than one type of failure to be considered. In such cases, a separate fault tree must be developed for each type. Examples would be:

(1) Loss of aircraft through control failure
(2) Loss of all aircraft position information
(3) Loss of Category II landing capability
(4) Loss of RNAV capability.

---

* Superscript numbers in parentheses refer to items in the Reference List.

This fundamental event to be studied is sometimes designated as the "Top Event", since it occurs at the top of the fault tree as usually drawn.

Once the top event has been selected, the next step is to enumerate all the ways in which the top event can happen. This enumeration is done through use of a specific type of graph structure known as a tree, hence the name fault tree.

If a given top event T can be caused by any one of the other events A, B or C, this can be depicted schematically as shown in Figure 2.

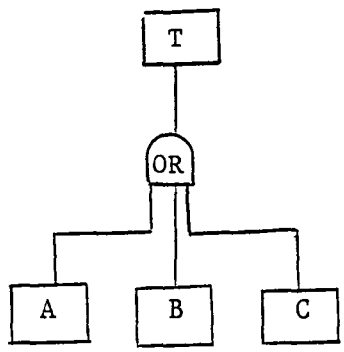


FIGURE 2. FAULT TREE WITH "OR" GATE

The notation of an "OR" gate is used to denote the fact that any one of the events A, B or C can cause T. The events A, B and C could be mutually exclusive or not. They could be statistically independent or not. Each of the events A, B, or C could be the top event in another fault tree. For example, there could be several other events which could cause A.

If T is caused by the presence of two or more events, the dependence is indicated as in Figure 3.

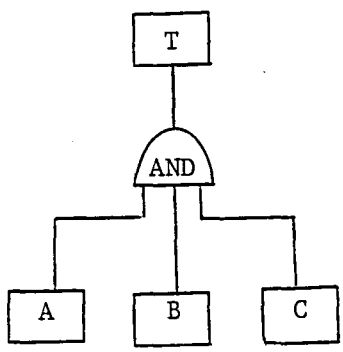


FIGURE 3. FAULT TREE WITH "AND" GATE

This means that all three events A, B and C must occur in order to cause T. Again, A, B or C could be the top event of another tree.

The construction of a complete fault tree proceeds from the top down. The top event is defined, and those events leading to the top event are defined. Then those events leading to the events just below the top event are defined. This continues until a level of fundamental events is reached. The nature of this fundamental level can be selected for the purposes of a particular problem. It could be failures of fundamental components such as resistors or solder joints. It could be failure of major subsystems, such as an inertial navigation unit or a particular computer function.

In concept, this is all there is to the construction of fault trees. In considering specific problems, however, considerable ingenuity may be required to fit the problem into this framework. For example, in the fault tree, there is no explicit recognition of time. This can be overcome, at least in many cases, by time-related definition of events. For example, a top event could be defined as loss of control during a specific period of time, such as final approach and landing. If there is more than one time period of interest, it may be necessary to construct a different fault tree for each time period, and for each top event in each time period. Conceptually, this is a simple approach, but the labor involved in constructing many fault trees could be considerable.

Also, it is necessary for the analyst to have a very good understanding of the system being analyzed. It is important that all ways of reaching the top event be portrayed in the tree. There is no general way to assure this, but the more the analyst knows about the working of the system, the less likely he is to overlook failure-producing events or combinations of events.

Determination of Failure Probabilities

Once a fault tree has been developed, it may be desirable to determine the probability of the top event. In order to do this, it is necessary to know the probabilities of the fundamental events. If the fundamental events

are independent, the determination of probabilities is relatively straightforward.

The situation of Figure 2 leads to the relationship

$$P_T = 1- (1-P_A)(1-P_B)(1-P_C)$$

where P_T is the probability of the top event, and P_A, P_B and P_C are the probabilities of the contributing events. If the probabilities are small, as is usually the case, this is well approximated by

$$P_T = P_A + P_B + P_C$$

In the situation of Figure 3, the probability is computed from

$$P_T = P_A P_B P_C$$

Probability computations start at the bottom of the tree with the fundamental events and proceed upward, using the above formulas at each stage until the top event is reached.

If events are not all independent, more complex computations may be required, but standard probability theory covers these cases. For problems of this type, special analytical methods and computer programs have been developed [13,14].

TASRA SYNOPSIS

General Discussion

The TASRA (Tabular System Reliability Analysis) model was developed by Battelle for performing reliability analyses of complex systems. It is well suited for this purpose in that the model can simulate real-world situations in which a malfunction occurs in the system with major portions of the system remaining operational, as well as a complete failure of the system. Most reliability models do not accommodate the malfunction situation readily.

The TASRA model used by Battelle to analyze and predict system and major assembly reliability is computer-based and configured so that the detailed functional inter-relationships of the subject system are represented

by the reliability model. Thus, failure of a subassembly or assembly in the real system will have the same effect on system operation as the reliability model depicts. If failure of one assembly causes a major system failure, the model will faithfully represent it. If failure of another assembly only degrades system operation, as determined by engineering analysis, the effect will be reflected in the probability of occurrence for that particular malfunction state without changing a related MTBF which is based on a failure state.

In a TASRA analysis, the term "malfunction" means a sometimes acceptable degradation in functional performance (e.g., three channels down out of five or transmitting at reduced power) while "failure" is used to indicate complete cessation of functional performance of the component or assembly (e.g., five channels down out of five). Thus, the failure of a subassembly could cause either a malfunction or failure of the next higher-level assembly depending on the functional interrelations between the two in the system. Such system-specific details can be represented by the TASRA reliability model used in this analysis. The model generates reliability data at each level of the system hierarchy, and for each failure state or defined malfunction state. These can be combined into a MTBF for a higher level if so desired.

Because of the operational realism TASRA offers, it can also be used as a tool to assist the system designers in achieving an improved trade-off between cost and reliability if desired. Early in the design/development cycle, the first iteration of the computer program will provide reliability predictions based on inputs of part failure rates or estimates of assembly reliability at the system level at which information is available. Given this initial information, the computer will predict a value of system reliability. If it is unacceptable, the computer outputs can be studied to identify those areas that need improvement to bring the MTBF up to an acceptable level. Changes in system design or reliability of the parts procured for particular assemblies can be evaluated to estimate the effects on the overall system reliability. In parallel with this, cost studies can be conducted to determine the impact of these changes on the cost of the system. Thus a TASRA analysis provides information that can be used in establishing the relationships between cost and reliability of a system.

Overview of TASRA Modeling Procedure

As Figure 4 shows, the user of the Tabular System Reliability Analysis (TASRA) model must generate a functional description of the total system, and of its subsystems, major assemblies, subassemblies, etc. The most important criteria in this step is to select "building-blocks" such that a failure of each is logically independent of the failure of the other building-blocks at that system level. A diagram is prepared to doucment this partitioning at each level. This level-by-level set of partitioned functional diagrams is one of the basic inputs the analyst must prepare to use the TASRA computer model. Input information from system designers knowledgeable in total system operation is usually necessary during this step.

Another concept essential to an understanding of the TASRA model is that of system states. The state of the system (from an operational reliability perspective) can be:

1) Fully operational, as the specifications define it, or
2) Failed (complete cessation of functional ability) -- called failure state, or
3) In one of several degraded operating modes -- called malfunction states.

The TASRA model can be used to predict the probability of occurrence of each state defined for each level of the system at which an analysis is conducted. This can be expressed as a mean time between failure (MTBF) or average time between occurrence (ATBO).

The analyst documents failure and malfunction state definitions working through the system level by level. Several iterations may be required to develop a consistent set of state definitions for each system level.

The decision portion of the analysis begins when the bottom of the procedural diagram of Figure 4 is reached. A bottom-up decision process of recording the system state that would occur as a consequence of each possible combination of 1-, 2-, and 3-at-a-time failures of the building-blocks for the system level under study is conducted. This is completed on standard tables developed by Battelle for this purpose.

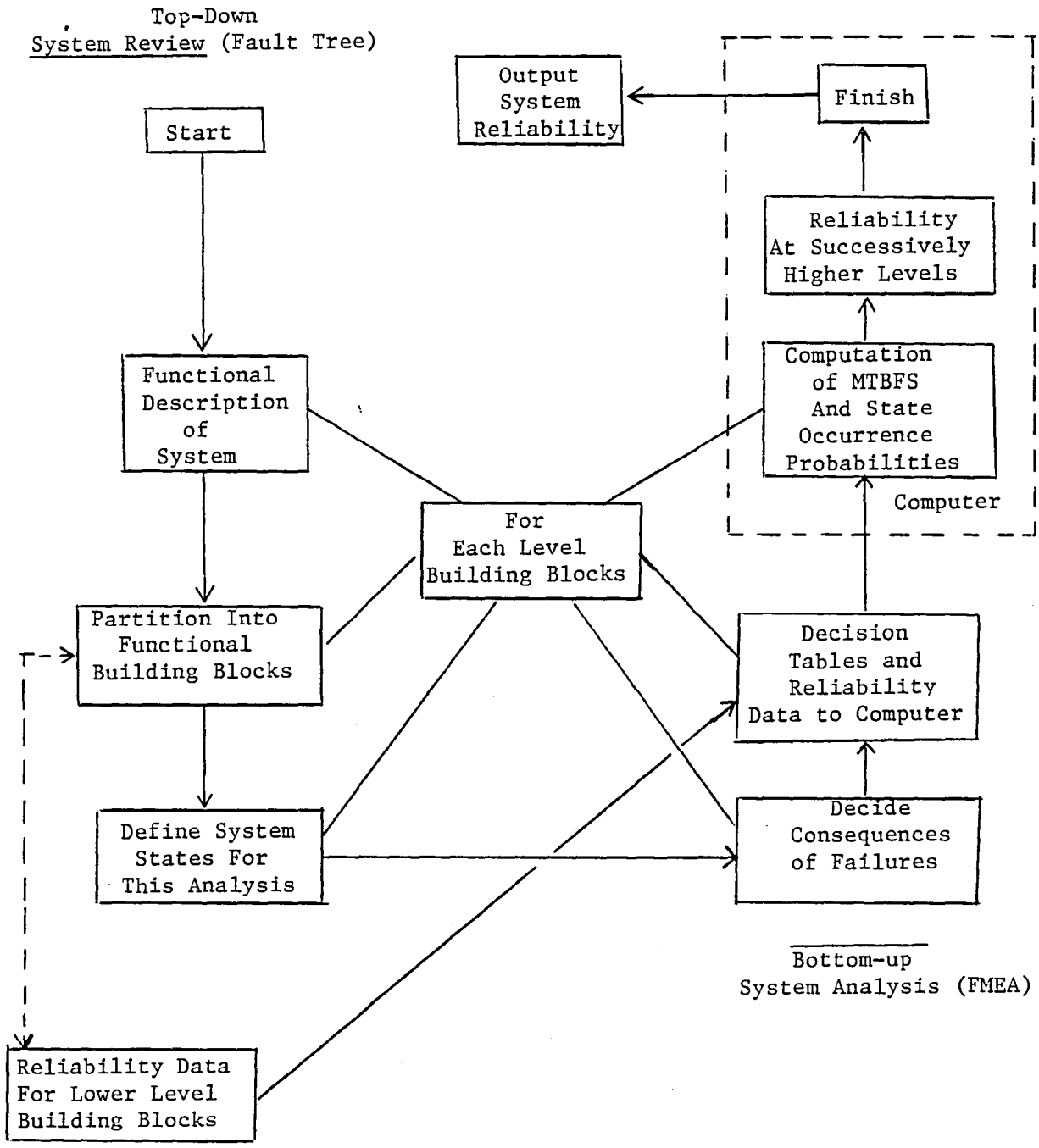


FIGURE 4. OVERVIEW OF PROCEDURE TO IMPLEMENT TASRA SYSTEM RELIABILITY MODEL

Figure 5 represents the flow of activities that take place at a given level within the system while carrying out the TASRA procedure. The activities on the right deal with functional partitioning, state definitions, and decisions and documenting of failure consequences. The activities on the left of the figure relate to reliability data inputs and when necessary, estimates of building-block reliability. Figure 6 then puts together the one-level activities of Figure 5 into the analysis of the entire system.

As illustrated in Figure 6, the procedure of Figure 5 is repeated at each level of the system until the analysis is completed up to the top level of the system hierarchy. At this point, one iteration of the TASRA system reliability model is complete, and reliability estimates (probabilities of state occurrence) are available for all of the failure and malfunction definitions at each system level. These probabilities may also be presented as MTBF's by the computer which is programmed to assume exponential distributions for this calculation. The calculations may be iterated as required to incorporate new data or changes in system structure.

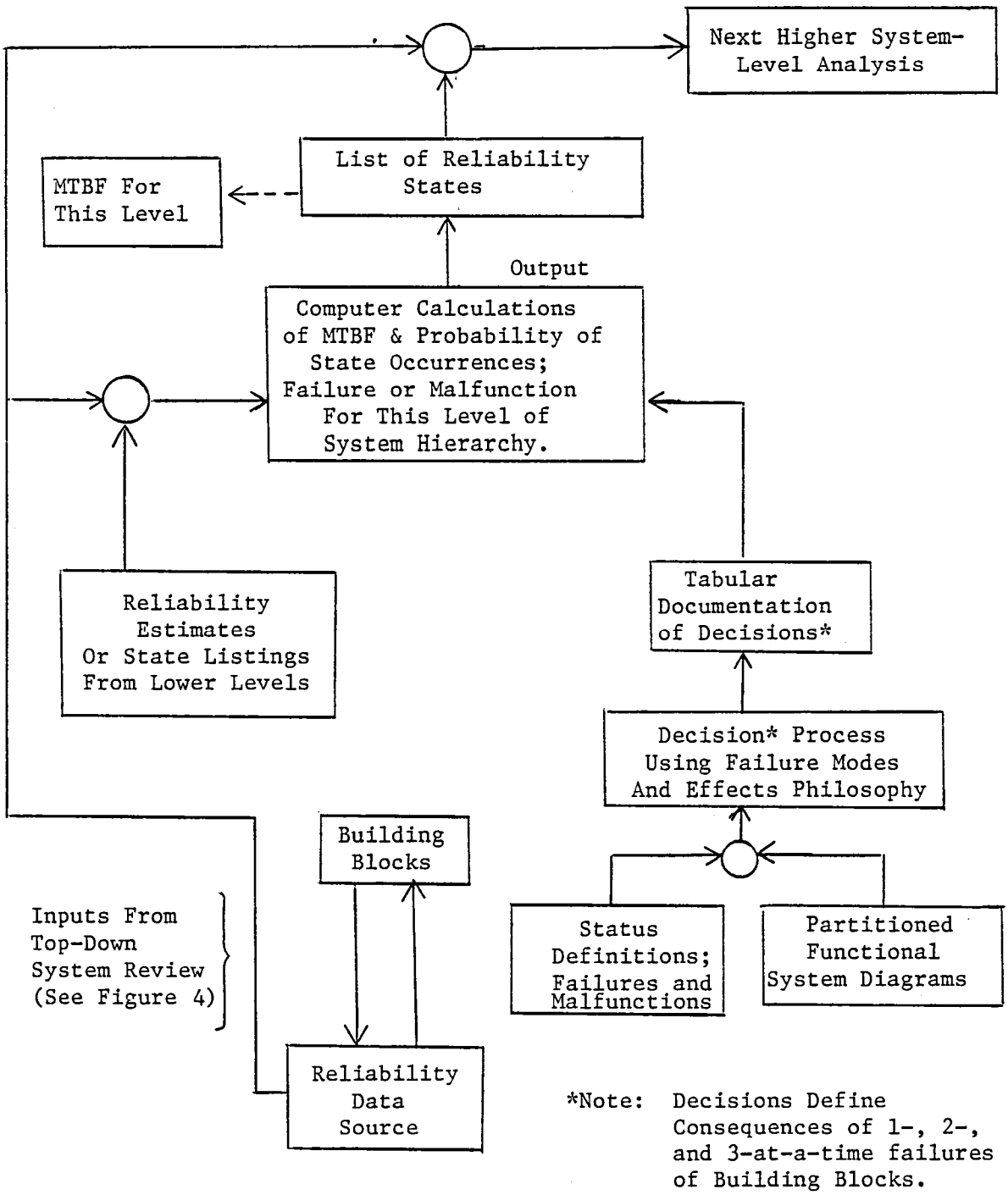


FIGURE 5. BASIC MODULAR PROCEDURE OF BOTTOM-UP SYSTEM RELIABILITY ANALYSIS AND PREDICTION (FOR ONE SYSTEM LEVEL)

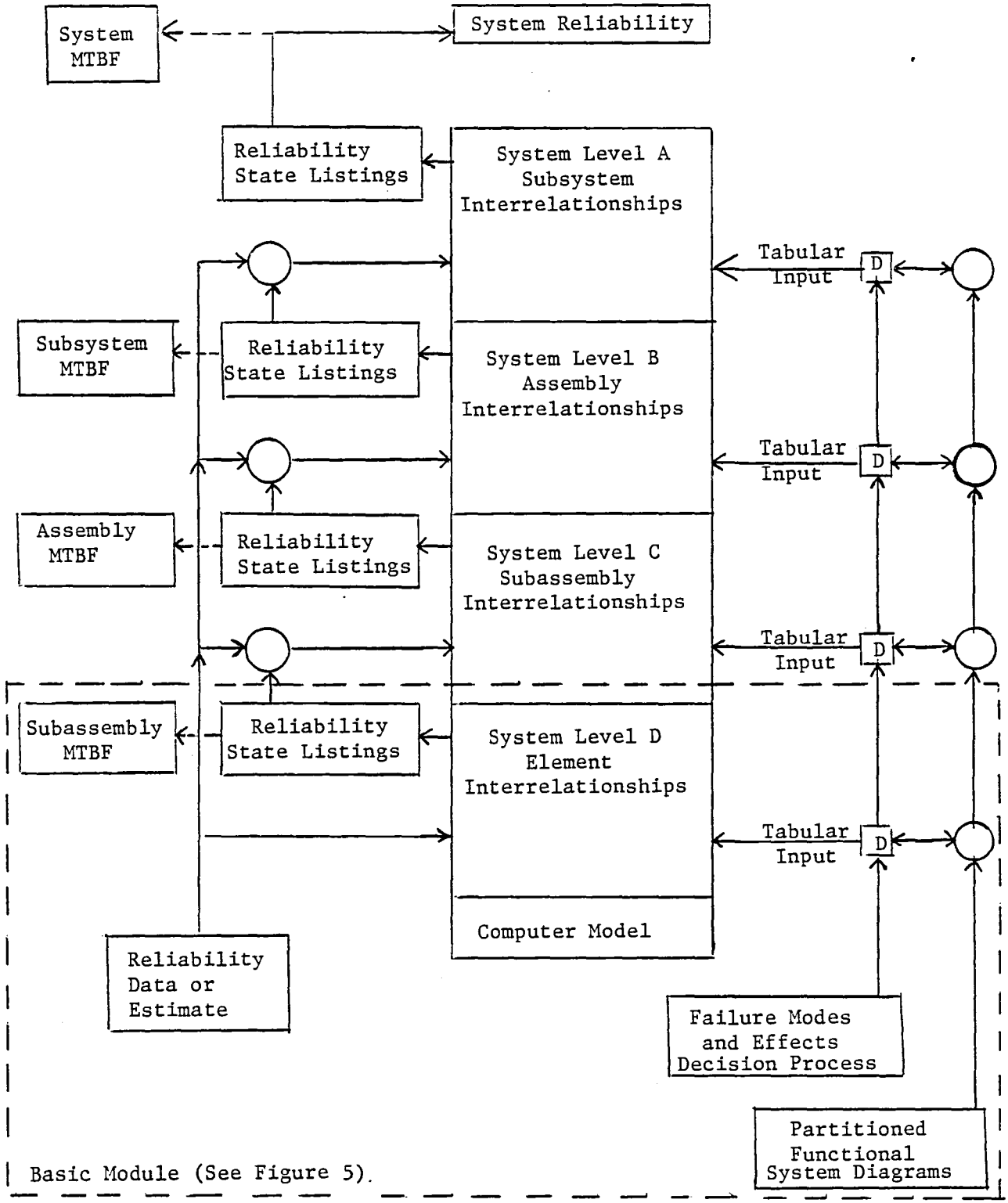


FIGURE 6. STRUCTURE OF SYSTEM RELIABILITY ANALYSIS MODEL

SCENARIOS AND FAULT TOLERANT SYSTEM DESIGNED AND ANALYZED

SERIES-PARALLEL DESIGN

A simple series-parallel problem, depicted in Figure 7 was the first problem analyzed to verify the researcher's understanding of the respective techniques and obtain a preliminary comparison of the relative ease with which the techniques could be applied to a simple problem not involving time or environmental dependency. The subsystems depicted in Figure 7 each have one failure mode and all subsystem failures are independent. There is no failure sensing for each of the subsystems and there is no possibility of repair. The failure of each subsystem is equivalent to that of an open circuit. Subsystems C and D are parallel redundant with branch operation of either assuring system success. Branch A-B is parallel redundant with branch C-D, that is either branch yields system success. Investigators were instructed to assume an exponential permit failure rate (Poisson distribution). The data for each subsystem is given in Table 1.

TABLE 1. SERIES-PARALLEL SUBSYSTEMS FAILURE RATES

Subsystem	λ
A	5×10^{-4}
B	4×10^{-4}
C	1×10^{-3}
D	1×10^{-3}

The analysts were to compute, for time equal to 10 hours, the probability of complete failure of the total system and the system reliability.

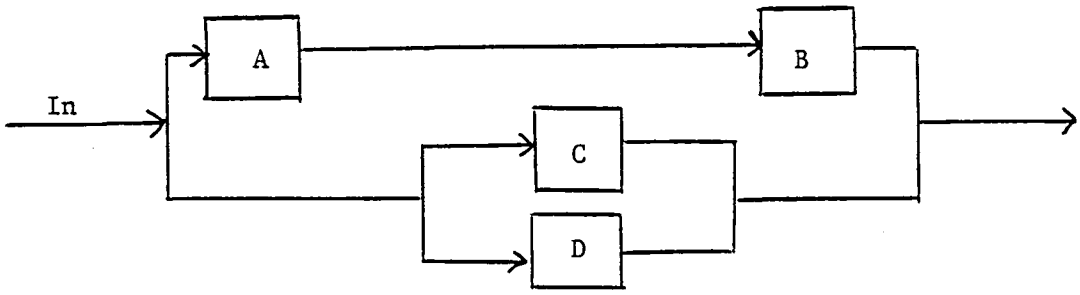


FIGURE 7. SERIES-PARALLEL DESIGN

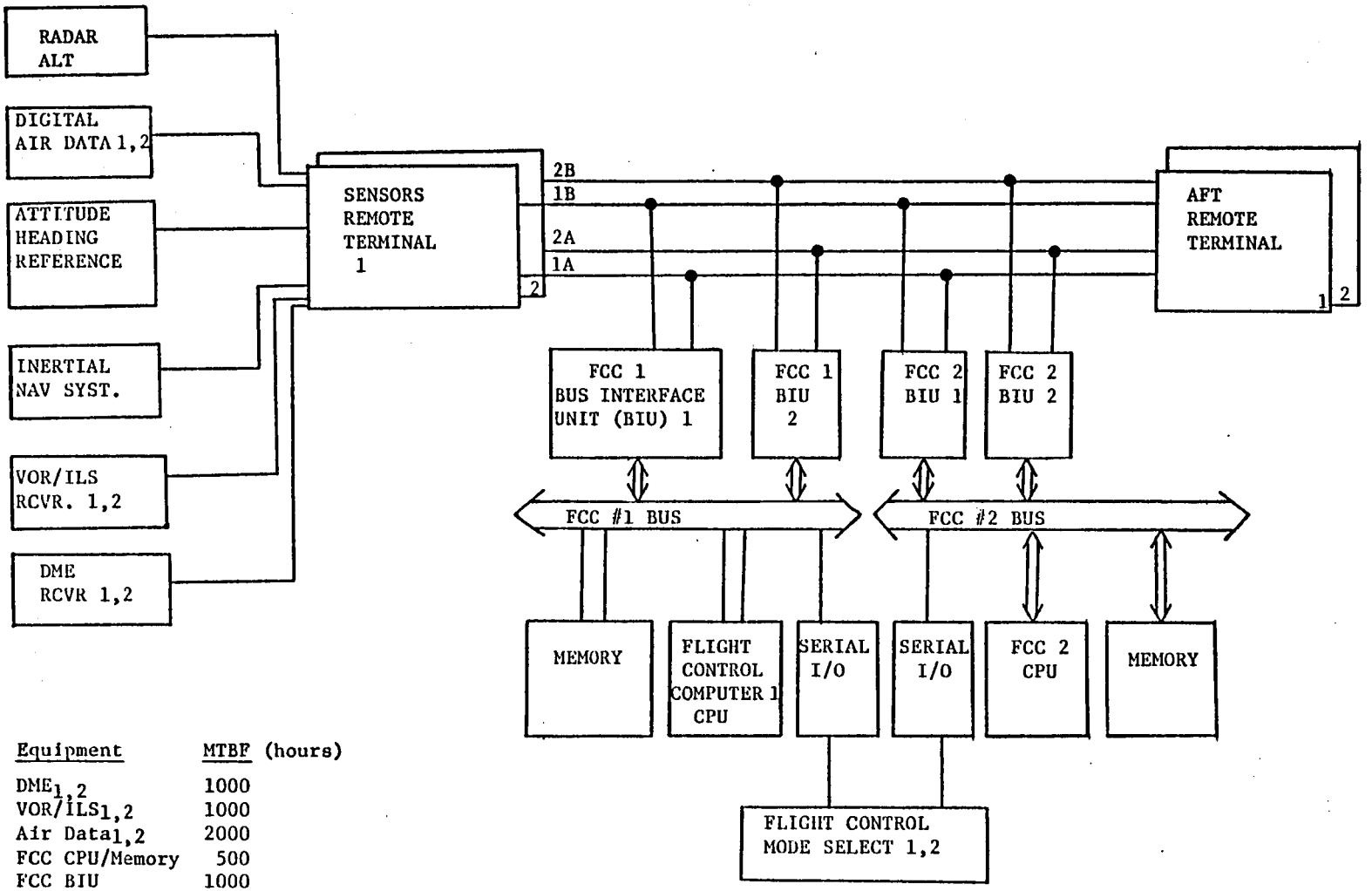
DUAL-DUAL SYSTEM

Figure 8 represents a portion of a digital flight control system which is dual-dual fail-operating. The servo amplifiers and monitor elements and servo sets connected to the actual sensors are not shown to keep the problem within bounds. The sensors are cross-strapped to two remote terminals which convert the sensor signals to digital signals which are tranmitted, on command, over one of the redundant busses for each remote terminal to the flight control computers.

The principal functions to be performed are the state estimation function and the command generation/execution function. Note that a single radar altimeter, attitude heading reference set, and inertial navigation system are carried. Dual-digital air data systems, VOR/ILS receivers and DME receivers are carried and input to both remote terminals. Each remote terminal terminal has a dual redundant bus which interfaces with a bus interface unit that interfaces with the flight control computer bus and hence flight control computer. The dual redundant data bus also interfaces with the remote terminal. In other words, aft remote terminal one and sensor remote terminal one have dual redundant busses 1A and 1B and aft remote terminal two and sensor remote terminal two have busses 2A and 2B. Flight control mode selection is redundant and interfaces with each of the flight control computers through a serial input/output panel.

Scenario

The mission consists of three phases. The first phase is a takeoff/climb phase and is fifteen minutes in duration. The second phase is a cruise phase of forty-five minutes duration. The descent and landing phase consists of fifteen minutes. Assume all equipment is operating at takeoff. During cruise, weather conditions at the scheduled destination develop requiring Category II capability. As stated in FAA Advisory Circular 120-29, Category II conditions require both ILS and glide slope receivers to be operable, the radar altimeter to be operable, both flight control computers to be operable, as well as an attitude reference source such as the attitude heading reference



Equipment	MTBF (hours)
$DME_{1,2}$	1000
$VOR/ILS_{1,2}$	1000
$Air\ Data_{1,2}$	2000
FCC CPU/Memory	500
FCC BIU	1000
INS	300
AHR	800
Radar Alt.	700
Remote Terminals	500
Flt. Cntrl Mode Select	2000

FIGURE 8. DUAL-DUAL SYSTEM

or inertial navigation system. Both digital air data systems must also be operable. Table 2 lists the components required for each of the mission phases.

For the purpose of the analysis, the final approach and touchdown phase lasts for two minutes.

Table 2 lists the equipment required for a safe flight, the equipment required to initiate the Category II landing at time equal to 73 minutes, and the equipment required to complete the Category II landing. The analysts calculated the probability of failure to initiate the landing and hence divert to the alternate airport due to loss of equipment required to initiate the landing, probability of successfully landing at the original destination, and probability of loss of the aircraft (unsafe flight) using the data in Figure 8 and Table 2.

At all times, each component is either totally operating or totally failed. The hardware and software associated with detecting component failures and removing failed elements is assumed to be perfectly accurate and perfectly reliable. Failures in each component have an exponential (Poisson) distribution. The Category II Approach and Landing can be aborted any time until T = 75 minutes.

MULTI-PROCESSOR SYSTEM

Scenario

The scenario for the third problem involved a mission consisting of five flight phases which are given in Table 3 with the corresponding duration of each phase and probabilistic weather at the destination at the time of scheduled departure.

The takeoff phase is assumed to start when the throttles are advanced to begin the takeoff roll after taking the active runway. The landing phase ends when the aircraft exits the active runway after decelerating to turnoff velocity. The weather at the destination is cloudy and the probability of the weather requiring Category II capability is 0.05 at the beginning of cruise.

System Design

The system configuration in Figure 9 represents a portion of a digital flight control system which possesses some of the features of the Fault Tolerant Multi-Processor (FTMP) architecture developed by C. S. Draper Laboratories.

TABLE 2. COMPONENT REQUIREMENTS FOR MISSION PERFORMANCE LEVELS

Component	MINIMUM COMPONENT REQUIREMENTS		
	Safe Flight (both phases)	Initiate CAT II Landing (T=73 min)	Complete CAT II Landing (T=75 min)
Radar Alt.		1	1
Digital Air Data	1	2	(1 and 1) or 1
AHRS	1 or 1	1 or 1	
INS			
VOR		2	1
DME		1	
Sensor RT	1	2	1
PU-I	1		1
PU-II		2	
FCMS	1	2	1
Aft RT	1	2	1

where

PU = processing unit
PU-I: one FCC with one associated BIU
PU-II: one FCC with both associated BIUs

TABLE 3. MISSION FLIGHT PROFILE

Flight Phase	Duration (minutes)
1. Takeoff	3
2. Climb-Out	8
3. Cruise	51
4. Let-Down	10
5. Landing	3

A quintuple redundant bus structure is employed with each of the five bus sets consisting of six lines. Two of the six lines in a bus set are dedicated to processor transmission (output) to common memory and registers; one line of the six is dedicated to common memory transmissions (output); one of the six is dedicated to clock generator transmission; one of the six is dedicated to I/O port input transmissions; and the last of the six is dedicated to I/O port output transmissions.

Each processor contains an independent processor-cache memory module, and common memory modules which communicate with other processors via the redundant serial busses. All information processing and transmission is conducted in triplicate by a triad of processors so that local voters in each module can detect errors. Each processor triad acts as one functional processor, of which several can work in parallel. The core software is assumed to handle fault detection, diagnosis, and recovery in such a way that applications programs do not need to be involved.

The procedures of each job reside in common memory. Each job step is scheduled to occur at a given time or following a given event. Relevant dispatch data for each scheduled job step is kept in a queue. Job assignments are all made on a floating basis, so that any available processor triad is eligible to execute any job step. When a processor fails, its triad will attempt to complete its current job step, which it will do unless a second failure occurs during the milliseconds required to complete the job step. When the job step is complete, one of the other processor triads is assigned

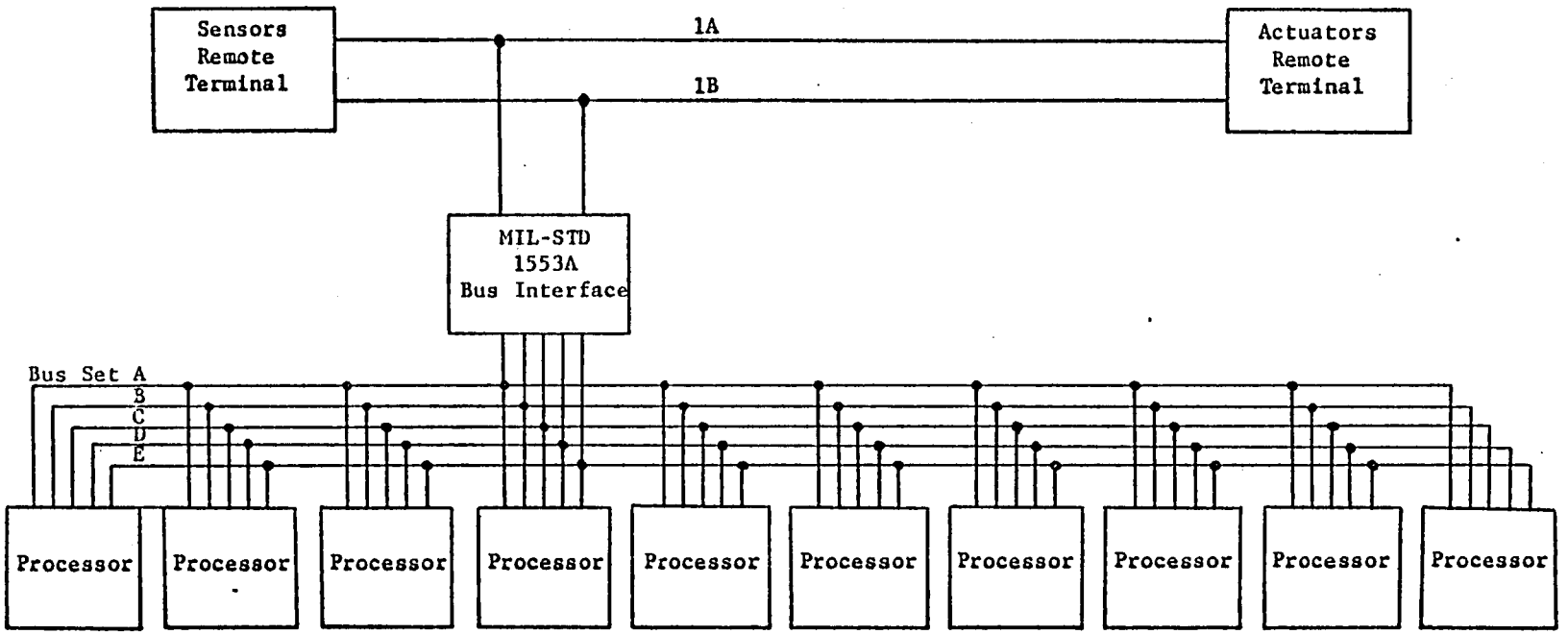


FIGURE 9. MULTI-PROCESSOR SYSTEM CONFIGURATION

the task of controlling the reconfiguration of the "injured" triad. Modules can be retired and/or reassigned in any configuration. Reconfiguration is carried out routinely from second to second to search for latent faults in the voting and reconfiguration elements.

The functions (and their priority) and subfunctions (tasks) to be performed are given in Table 4. The priority of the tasks (and associated job steps) is used by the processor triads in their selection (from common memory) of the next job step to be executed. For the purpose of this problem, only the functions' priority was considered. The functions' priority and criticality correspond to those given in Table 5.

The sensors interfacing with the sensor remote terminal, and servo amplifiers, monitor elements, and servo sets (connected to the actual actuators) interfacing with the actuators remote terminal were not considered to keep the problem within bounds. In working this problem, it was assumed that:

(1) All processors, remote terminals, and the MIL-STD-1553A bus interface are properly functioning at the beginning of the take-off phase, t = 0.

(2) Only permanent failures need be considered; that is, each component is either totally failed or totally operating.

(3) All components are nominally required to function during the entire flight.

(4) Fault detection and reconfiguration are assumed to be perfectly accurate and no second failures occur during reconfiguration (Admittedly, this is a bad assumption since a finite amount of time is required for reconfiguration but the problem analysis objective does not suffer from this assumption).

(5) All bus sets and the MIL-STD-1553A dual bus (itself) are perfectly reliable.

(6) The Flight Management function is required from t = 0 to t = 72, i.e. for Phases 1-4 in order to arrive on time.

(7) The remote terminals and the MIL-STD-1553A bus interface with the MIL-STD-1553A busses have redundant input/output (I/O) channels A and B with equal reliability. In order for data transfer and hence safe flight to be successful, the conditions in Table 6 must be satisfied. In other words, a sensor-to-bus interface and a bus interface to actuator channel must exist for data transfer and hence safe flight.

TABLE 4. AUTOMATIC FLIGHT FUNCTIONS

Priority	Function	Subfunction
3	Flight Augmentation Control	1. Artificial Feel 2. Pitch Trim 3. Stability Augmentation a. Mach/IAS Augmentation b. Pitch Augmentation c. Wing-Load Alleviation Augmentation d. Flutter Suppression Augmentation e. Ride Control Augmentation f. Roll Augmentation g. Yaw Augmentation 4. Rudder Ratio Changer 5. Direct Lift Control 6. Aileron Gain Programming 7. Flap Limiting
2	Flight Control	Attitude Hold Heading Hold Control Wheel Steering Altitude Hold Automatic Approach and Landing Autothrottle Air Speed Select Air Speed Hold Missed Approach Back Course Localizer Flight Director Signals Heading Select Course Select Flight Envelope Protection
1	Flight Management	Performance Management Lateral Navigation and Guidance Heading Select/Hold Course Select/Hold

TABLE 4. (Continued)

Priority	Function	Subfunction
1	Flight Management (continued)	Vertical Navigation and Guidance Vertical Speed Select/Hold Altitude Select/Hold Thrust Axis Control Airspeed/Mach Hold Airspeed/Mach Select 4-D Guidance Electronic Flight Instrument System Management Data Update Interface Inertial Reference System Initialization and Heading Set

TABLE 5. FUNCTION PRIORITY/CRITICALITY

Function	Priority	Criticality
Flight Augmentation	3	Required for Safe Flight. As failures occur, it is always given priority over Flight Control and Management. After spares are depleted, assume loss of one processor results in loss of the aircraft.
Flight Control	2	Required to initiate and complete CATEGORY II or III approach and landing. Loss results in reduced operative performance and increases pilot workload. It is given priority over Flight Management. Not required for dispatch unless adverse weather expected during flight.
Flight Management	1	Required for energy efficient and on-time flight. Loss results in flying radials between VORTACS and extends flying time. Not required for dispatch.

TABLE 6. MIL-STD-1553A DATA TRANSFER LOGIC

Sensor Channel		MIL-STD-1553A Bus Interface Channel		Actuator Channel		Safe Flight
A	B	A	B	A	B	
Y	Y	Y	Y	Y	Y	Y
Y	Y	Y	Y	Y	N	Y
Y	Y	Y	Y	N	Y	Y
Y	Y	Y	Y	N	N	N
Y	Y	Y	N	Y	Y	Y
Y	Y	Y	N	Y	N	Y
Y	Y	Y	N	N	Y	N
Y	Y	Y	N	N	N	N
Y	Y	N	Y	Y	Y	Y
Y	Y	N	Y	Y	N	N
Y	Y	N	Y	N	Y	Y
Y	Y	N	Y	N	N	N
Y	Y	N	N	:	:	N
:	:	:	:	:	:	N
Y	N	Y	Y	Y	Y	Y
Y	N	Y	Y	Y	N	Y
Y	N	Y	Y	N	Y	Y
Y	N	Y	Y	N	N	N
Y	N	Y	N	Y	Y	Y
Y	N	Y	N	Y	N	Y
Y	N	Y	N	N	Y	N
Y	N	Y	N	N	N	N
Y	N	N	Y	Y	Y	N
:	:	:	:	:	:	N
N	Y	Y	Y	Y	Y	Y
N	Y	Y	Y	Y	N	Y
N	Y	Y	Y	N	Y	Y
N	Y	Y	Y	N	N	N
N	Y	Y	N	Y	Y	N
:	:	:	:	:	:	N
N	Y	N	Y	Y	Y	Y
N	Y	N	Y	Y	N	N
N	Y	N	Y	N	Y	Y

(8) Each triad performs a single function as shown in Table 7. With less than three processors out of ten functioning, the aircraft is assumed to crash.

(9) The subsystems permanent failure rates are constant (exponential model). The reciprocals of the failure rates (MTBF) are given in Table 8.

The analysts computed the following mission outcomes:

(1) Probability of successful on-time landing at the original destination.

(2) Probability of successful, but late, (based on flight management loss prior to landing phase) landing at the original destination and the expected economic penalty for inefficient flight (see Table 9).

(3) Probability of diverting and safely landing at the alternate destination. Diversion only occurs during phases 3, 4, and 5 if CAT II capability is required and not available. The expected economic penalty was computed using Table 9 data. It was assumed that the flight time to the alternate is the same as the remaining flight time to the original destination.

(4) Probability of aborting (due to loss of all spares with only a triad remaining and safely landing at the origin during phases 1 and 2. It was assumed that abort at end of phase 1 transitions to phase 5 (with VFR); abort at end of phase 2 transitions to phase 4 followed by phase 5 (with VFR).

(5) Probability of loss of aircraft during the mission; and, probability of loss of aircraft during each phase.

MULTI-PROCESSOR DESIGN MODIFIED FOR CROSS-TRAINING

The system design used for the cross-training of the analysts on the respective fault tree and "performability analysis" methods was a variation of the multi-processor design previously analyzed. For this case, the same design applies but the reliability of the remote terminals and bus interface units was assumed to be perfect. Only the reliability of the ten processors was considered.

TABLE 7. PROCESSORS REQUIRED FOR FLIGHT FUNCTIONS

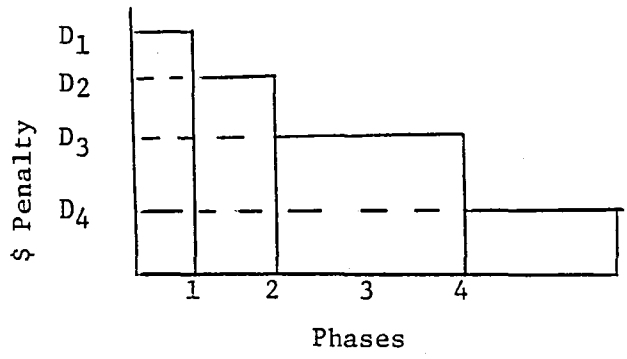
Processors			Triads	Flight Functions Operating		
Functioning	Failed	Spare		Augmentation	Control	Management
10	0	1	3	Y	Y	Y
9	1	0	3	Y	Y	Y
8	2	2	2	Y	Y	N
7	3	1	2	Y	Y	N
6	4	0	2	Y	Y	N
5	5	2	1	Y	N	N
4	6	1	1	Y	N	N
3	7	0	1	Y	N	N
2	8	0	0	N = Land Immediately		
1	9	0	0	N = Crash		
0	10	0	0	Crash		

TABLE 8. SUBSYSTEM DATA

Subsystem	MTBF (Hours)
Processor	100
Bus Interface Channel A	500
Bus Interface Channel B	500
Remote Terminals Channel A	500
Remote Terminals Channel B	500

TABLE 9. ECONOMICS PENALTIES DATA

1. Inefficient Flight (i.e., Loss of Flight Management [FM])



Loss of FM during phase j (but not prior to phase j) causes an economic penalty of $\$D_j$.

2. Diversion and Safe Flight

D_5 = $ penalty

Assume $D_5 > 10. D_1$

3. Abort and Safe Return to Point of Origination

D_6 = $ Penalty

The same scenarios used for the multi-processor problem was used in the cross-training problem. The analysts were instructed to compute the probability of safe on-time arrival at the original destination. If time and funds permitted, the analysts were also allowed to compute the probability of successful late landing at the original destination, probability of diversion with a safe landing, and the probability of loss of the aircraft.

ANALYSIS RESULTS

SERIES-PARALLEL PROBLEM

The series-parallel problem was an elementary problem used primarily for learning purposes by the analysts concerned with fault trees and performability analysis. Solutions using those two techniques were quickly and easily obtained; they were numerically equal. One man-hour was expended for performability analysis and one and one-half man-hours were used for the fault-tree analysis.

The TASRA solution was also quickly obtained, but it was not equivalent to the other two solutions. A model error which had the net effect of interchanging failure rates among components was located. Following correction of the error, the TASRA results agreed with the other results. One man-hour and nine system seconds of computer time were expended on the TASRA analysis and documentation.

DUAL-DUAL SYSTEM ANALYSIS

Summary of Results

Table 10 summarizes the results of applying the three analysis techniques to the previously described dual-dual system. Numerical results for performability analysis and the fault-tree approach are in close agreement. The TASRA values exhibit some disparities compared to the other values. The differences apparently are caused by the procedure used to combine components into subsystems and then the system, since it assumes the aggregated entities will have exponential failure distributions. The man-hour figures are for problem formulation and solution; they do not include time to check the results to resolve numerical differences between techniques.

A summary of the performability analysis solution is given in the next subsection. Details are provided in Appendix A. The fault-tree analysis follows the performability solution. The TASRA solution is then described.

TABLE 10. DUAL-DUAL SYSTEM ANALYSIS RESULTS FOR THE THREE TECHNIQUES

	Performability	Fault Trees	TASRA
Mission Outcome Probabilities			
Safe Flight and Landing at Primary Destination	0.974212	0.974245	0.974236
Safe Flight and Landing at Alternate Destination	0.025763	0.025701	0.025740
Loss of Aircraft	25.98×10^{-6}	25.98×10^{-6}	23.69×10^{-6}
Man-Hours for Solution	46	30	25

Performability Analysis Solution

Analytic Summary

The performability analysis solution of the dual-dual system problem used three model levels--mission, function, and component--in addition to the accomplishment set. A concept of "independence with respect to mission outcomes" was used to accommodate the large number of trajectories in the base (i.e., component level) model. Probability computations were performed using the matrix multiplication procedures of performability analysis. The following paragraphs summarize the models and computations used to analyze the dual-dual system; details are provided in Appendix A.

The accomplishment set is $A = \{a_0, a_1, a_2\}$ where the a_i represent specific mission outcomes (i.e., accomplishment levels) of interest. In particular,

- a_0 represents safe flight and successful landing at the primary destination/
- a_1 represents safe flight and successful landing at the alternate destination;
- a_2 represents loss of the aircraft (unsafe flight or unsuccessful landing).

The base model is defined in terms of thirteen component types used in the dual-dual system and is also called the component level, or level 2, model. Two phases of the mission are used. Phase 1 is the time from takeoff (t = 9 minutes) to initiation of landing (t = 73 minutes). Phase 2 is the time from initiation of landing (t = 73 minutes) to completion of landing (t = 75 minutes). The specific variables used are x_{ij}, the number of units of component type i (i = 1, 2, ..., 13) which are fault-free for phase j (j = 1, 2). Each base model trajectory is represented by a 13-by-2 matrix in which rows correspond to component types and columns correspond to phases. Each such trajectory corresponds to a single accomplishment level.

As described in the synopsis of performability analysis, the first step is to determine the set of base model trajectories which results in the mission outcome a_i for every a_i in A. Two model levels--mission and function--were used to form the logical connection between the base model and the accomplishment set.

The mission level (level 0) model consists of two binary variables h_1 and h_2, representing the conditions required for "no diversion" and for "safe flight", respectively. Each mission level trajectory is of the form $\begin{bmatrix} h_1 \\ h_2 \end{bmatrix}$. The set of mission level trajectories corresponding to each a_i and A was determined directly from the definitions of the a_i and of h_1 and h_2. These inverses are:

$$\gamma_0^{-1}(a_0) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$\gamma_0^{-1}(a_1) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$\gamma_0^{-1}(a_2) = \begin{bmatrix} * \\ 1 \end{bmatrix}$$

where * indicates "any possible value" (in this case, 0 or 1).

The function level (level 1) model consists of four variables (f_i, i = 1, 2, 3, 4), one for each function. A function is defined as the set of jobs performed by a group of components. Groups are comprised of components which are related in some way. For example, the digital air data, attitude heading reference system, and inertial navigation system have interacting roles during the mission. Each function variable is defined as follows:

$$f_i = \begin{cases} 2 \text{ if function i meets the "no diversion" and "safe flight" requirements;} \\ 1 \text{ if function i meets the "safe flight" but not the "no diversion" requirements;} \\ 0 \text{ otherwise.} \end{cases}$$

A function level trajectory is then a column vector of the form

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

The inverse image of each mission level trajectory in the function level trajectory space was determined using the process described in the synopsis of performability analysis. Details are given in Appendix A. The function level inverses of the accomplishment levels are shown in Table 11.

TABLE 11. FUNCTION LEVEL INVERSES OF THE ACCOMPLISHMENT LEVELS

Accomplishment Level, a_i	Function Level Inverses, $\gamma_1^{-1}(a_i)$
a_0 (safe, no diversion)	$\begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \end{bmatrix}$
a_1 (safe, diversion)	$\begin{bmatrix} 1 \\ 1 \text{ or } 2 \\ 1 \text{ or } 2 \\ 1 \text{ or } 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ 1 \text{ or } 2 \\ 1 \text{ or } 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \\ 1 \\ 1 \text{ or } 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \\ 2 \\ 1 \end{bmatrix}$
a_2 (unsafe)	$\begin{bmatrix} 0 \\ * \\ * \\ * \end{bmatrix}, \begin{bmatrix} 1 \text{ or } 2 \\ 0 \\ * \\ * \end{bmatrix} \begin{bmatrix} 1 \text{ or } 2 \\ 1 \text{ or } 2 \\ 0 \\ * \end{bmatrix} \begin{bmatrix} 1 \text{ or } 2 \\ 1 \text{ or } 2 \\ 1 \text{ or } 2 \\ 0 \end{bmatrix}$

"*" represents "any possible value" (i.e., 0, 1 or 2)

The process of determining all base model trajectories which map to a given function level trajectory (i.e., the inverse image of the function level trajectory) is detailed in Appendix A. Basically, the approach is to find the inverse image for each component of the function level trajectory and then form the intersection of those images.

A practical problem encountered at this point was the large number (4×10^{10}) of mathematically possible base level trajectories. Every such trajectory must appear exactly once in the complete group of inverse image sets. Some method of writing many matrices in a reasonable amount of time was needed. Use of Cartesian sets* allows for efficient representation of sets. Notational conveniences such as using "*" to represent "any possible value" provide some limited help. Use of these approaches would still leave a burdensome task. The approach which relieves the burden is to take advantage of the mutual independence of groups of components. For example, the effect of the processors and bus interface units on the mission outcome is independent of the effect of the radar altimeter, VOR, and DME.

The concept of independence with respect to mission outcomes was used to divide the thirteen component types into four groups, each of whose trajectories were individually analyzed. The functions used in the previous model were chosen to correspond exactly with the independent component groups of the base model. A separate state diagram was then created for each component group.

Probability computations, the second step of performability analysis, were made using the four component groups. For each group, the computational procedure used the intraphase transition matrices, characteristic matrices, and vectors as described in the synopsis of the technique and References 1, 2, 3, and 9. The mission outcome probabilities for the four component groups were then combined in a straightforward way to determine the probability of each accomplishment level. An HP-25 hand calculator with eleven significant digits was used for the computations. The performability analysis results are:

Pr (safe flight and no diversion) = 0.974212

Pr (safe flight and diversion) = 0.025763

Pr (aircraft is lost) = 25.98×10^{-6}

---

* A set V is Cartesian if $V = R_1 \times R_2 \times \ldots \times R_k$ where R_i represents the set of all projections of elements of V onto their i^{th} coordinates.

Solution Effort

Application of performability analysis to the dual-dual system required a total of 46 man-hours. Of this total, 38 man-hours were used to formulate the model hierarchy, determine the inverse images (γ^{-1} (a_i)) of the accomplishment levels, and perform the probability computations. Another 8 man-hours were used to check the probability computations. Because of numerical discrepancies among the three techniques, the entire solution was then checked with an expenditure of 18 man-hours.

Discussion

Several difficulties were encountered in this application of performability analysis. The first significant problem was defining the model hierarchy. Both the accomplishment set and the base level model were readily defined using the problem statement. However, it was not clear how to define intermediate models to logically connect these two views of the system.

The large number of distinct component types (13 in the base model) result in over 10^{10} mathematically possible trajectories. Some method of decomposing this large state set was obviously needed. This was the motivation for dividing the base model into four component groups which were mutually independent with respect to their effects on the mission outcome. These groups provided the basis for defining four "functions" and the function level model.

The mission level model, which lies between the function level and the accomplishment set, was straightforward to define and use. While the mission level model could have been omitted from this problem, its use provides a better representation of performability analysis.

One conceptual error was made in determining groups of components which were independent with respect to their effects on the mission outcome. The error was an oversight regarding a dependency involving five components from two different groups and a landing requiring Category II weather capability. As shown in Appendix A, the probability of the event representing the error is on the order of 10^{-14}. Since the error was quite small, the probability computations were not changed.

It should be noted that the error was not a result of the performability analysis technique. The error can be attributed to the complexity of the problem and the analyst's attempts to decompose it into manageable pieces. A supplementary analysis using the five components was performed to satisfy the analyst that the problem could have been formulated in a manner to capture the dependency. The associated state diagram involved 72 states. The associated state transition matrix would have been tedious to complete and use, but it would not have required any ingenuity on the part of the user.

Fault Tree Solution

Analytic Summary

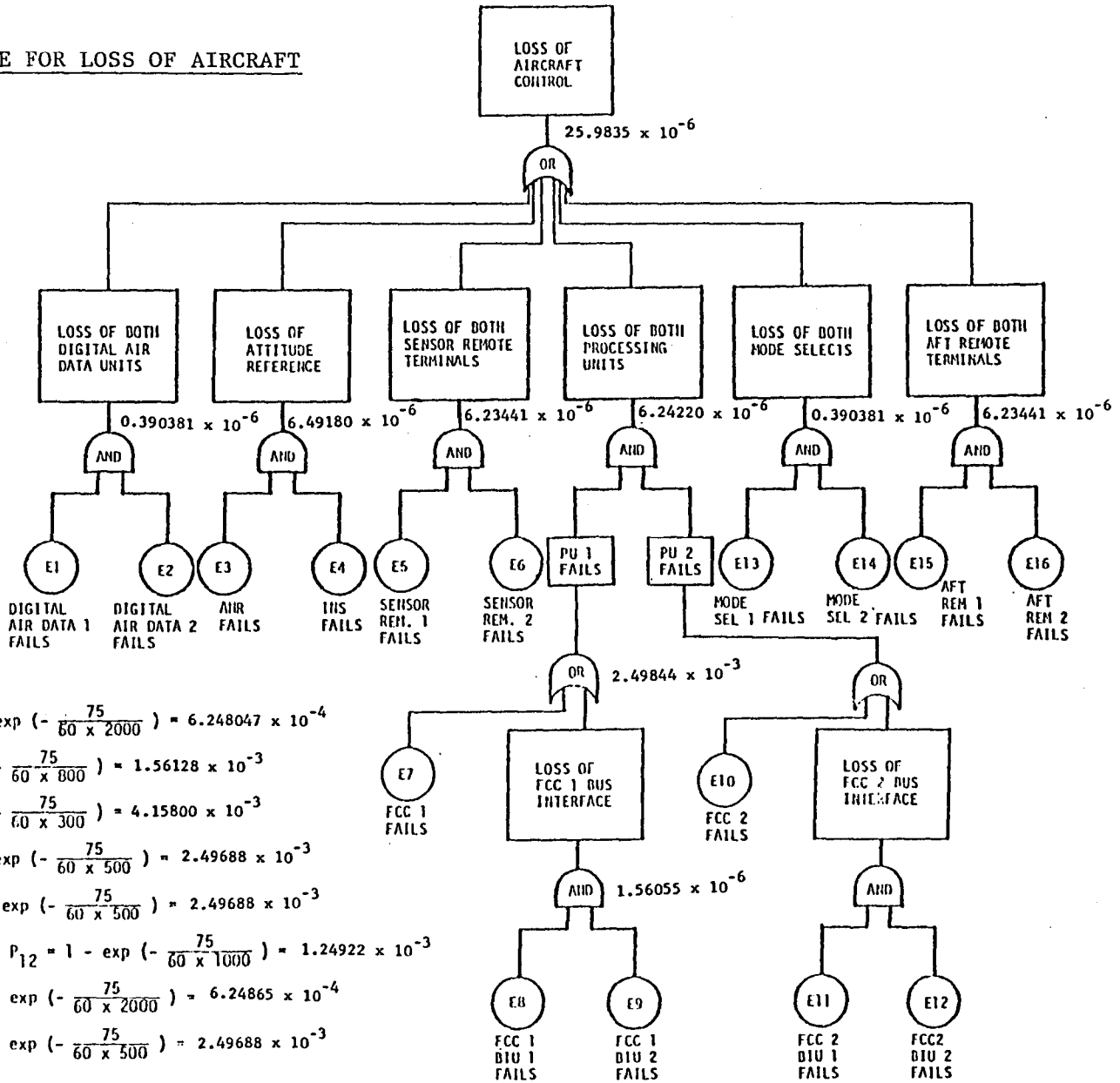
In this problem, three different probabilities are required. Accordingly, three different fault trees must be prepared.

Loss of Aircraft Control. The fault tree for loss of control is shown in Figure 10.

Failure to Initiate Cat II Landing. The fault tree for this case is shown in Figure 11.

Treatment of the Landing Phase. The landing phase differs from the earlier phase in that the operating complement of equipment is not uniquely defined at the start. In order to initiate the Cat II landing, all components must be operating with the following exceptions: (a) either one or two DME receivers, (b) either the AHRS, the INS, or both. Since the DME receiver is not involved in the landing phase, the question of whether one or both were operating at the start does not affect landing probabilities. Heading reference is needed during the landing, however, and the probability of completing the landing will definitely depend on which of the attitude equipments are operating at the start of the landing.

FIGURE 10. FAULT TREE FOR LOSS OF AIRCRAFT



$P_1 = P_2 = 1 - \exp\left(-\frac{75}{60 \times 2000}\right) = 6.248047 \times 10^{-4}$

$P_3 = 1 - \exp\left(-\frac{75}{60 \times 800}\right) = 1.56128 \times 10^{-3}$

$P_4 = 1 - \exp\left(-\frac{75}{60 \times 300}\right) = 4.15800 \times 10^{-3}$

$P_5 = P_6 = 1 - \exp\left(-\frac{75}{60 \times 500}\right) = 2.49688 \times 10^{-3}$

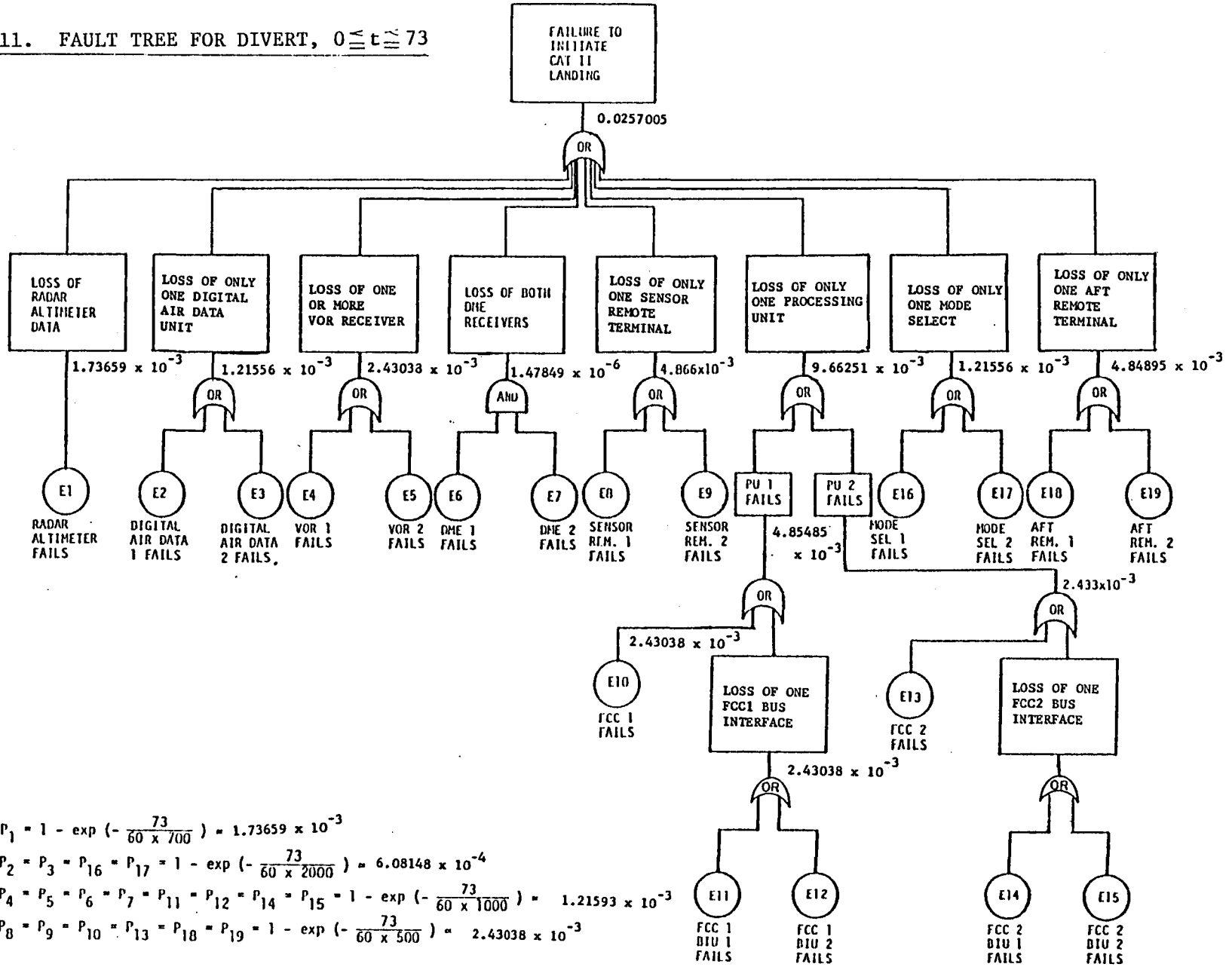
$P_7 = P_{10} = 1 - \exp\left(-\frac{75}{60 \times 500}\right) = 2.49688 \times 10^{-3}$

$P_8 = P_9 = P_{11} = P_{12} = 1 - \exp\left(-\frac{75}{60 \times 1000}\right) = 1.24922 \times 10^{-3}$

$P_{13} = P_{14} = 1 - \exp\left(-\frac{75}{60 \times 2000}\right) = 6.24865 \times 10^{-4}$

$P_{15} = P_{16} = 1 - \exp\left(-\frac{75}{60 \times 500}\right) = 2.49688 \times 10^{-3}$

FIGURE 11. FAULT TREE FOR DIVERT, $0 \leq t \leq 73$



$$P_1 = 1 - \exp\left(-\frac{73}{60 \times 700}\right) = 1.73659 \times 10^{-3}$$

$$P_2 = P_3 = P_{16} = P_{17} = 1 - \exp\left(-\frac{73}{60 \times 2000}\right) = 6.08148 \times 10^{-4}$$

$$P_4 = P_5 = P_6 = P_7 = P_{11} = P_{12} = P_{14} = P_{15} = 1 - \exp\left(-\frac{73}{60 \times 1000}\right) = 1.21593 \times 10^{-3}$$

$$P_8 = P_9 = P_{10} = P_{13} = P_{18} = P_{19} = 1 - \exp\left(-\frac{73}{60 \times 500}\right) = 2.43038 \times 10^{-3}$$

The following definitions will be used to develop the computational logic:

E_2 = successful initiation of Cat II landing
E_f = successful completion of Cat II landing
E_{AI} = both AHRS and INS operating at landing initiation
E_A = AHRS operating at landing initiation, but INS has failed
E_I = INS operating at landing initiation, but AHRS has failed
E_R = meeting all the non-heading requirements for landing initiation.

The principal task is, of course, to compute the probability of a successful landing (i.e., to determine $Pr[E_f]$). There are three and only three starting conditions for the landing: (1) E_R and E_{AI}, (2) E_R and E_A, and (3) E_R and E_I. It follows that

$$Pr[E_f] = Pr[E_R \text{ and } E_{AI}]Pr[E_f|E_R \text{ and } E_{AI}] + Pr[E_R \text{ and } E_A]Pr[E_f|E_R \text{ and } E_A] + Pr[E_R \text{ and } E_I]Pr[E_f|E_R \text{ and } E_I]$$

Since E_R and E_{AI} are associated with different equipment, they are independent events. It follows that

$$Pr[E_R \text{ and } E_{AI}] = Pr[E_R]Pr[E_{AI}]$$

Similar arguments can be made for the other two terms so that

$$Pr[E_f] = Pr[E_R]\{Pr[E_{AI}]Pr[E_f|E_R \text{ and } E_{AI}] + Pr[E_A]Pr[E_f|E_R \text{ and } E_A] + Pr[E_I]Pr[E_f|E_R \text{ and } E_I]\}$$

It can be seen from Figure 2 that $Pr[E_R] = 0.974299$. Also:

$$Pr[E_{AI}] = \exp\{\frac{-73}{60x800}\}\exp\{\frac{-73}{60x300}\} = 0.994438$$

$$Pr[E_A] = \exp\{\frac{-73}{60x800}\}[1-\exp\{\frac{-73}{60x300}\}] = 4.04115x10^{-3}$$

$$Pr[E_I] = [1-\exp\{\frac{-73}{60x800}\}]\exp\{\frac{-73}{60x300}\} = 1.51353x10^{-3}$$

The conditional failure probabilities are developed in Figures 12, 13, and 14. Substituting the results into the above equation for $Pr[E_f]$,

$$Pr[E_f] = 0.974299\{0.994438\ (1-4.76372\times10^{-5}) + 4.04115\times10^{-3}\ (1-8.92986\times10^{-5}) + 1.51353\times10^{-3}\ (1-15.8738\times10^{-5})\}$$

$$= 0.974299\{0.994391 + 4.04079\times10^{-3} + 1.51329\times10^{-3}\}$$

$$= 0.974245$$

Solution Effort

Determination of the time required for this problem is somewhat difficult because the problem went through some re-definition after it was initially stated. In addition, some time was lost in the solution by an erroneous interpretation of the problem statement which was finally developed. It is estimated that some 30 hours would have been required had these problems not been present. Actually, some 45 hours were spent on all activities associated with this problem.

TASRA Solution

Analytic Summary

Figure 15 depicts the reliability block diagram of the dual-dual system used for constructing TASRA inputs. Each assembly was given a set of identifying numbers to uniquely reference its possible states. For example, assembly 20 was the sensor terminals. The numbers 20.0, 20.1, and 20.2 refer to the states "both sensor terminals fault free", "both terminals failed", and "one terminal failed", respectively. Except for the fundamental assemblies, each assembly consists of a number of subassemblies. A logic table was created for each such assembly to specify the assembly state resulting from each possible combination of subassembly states. Finally, the failure rate data were input.

Two runs of TASRA were used to derive the numerical results. The first run corresponded to the first phase (i.e., takeoff to initiate landing).

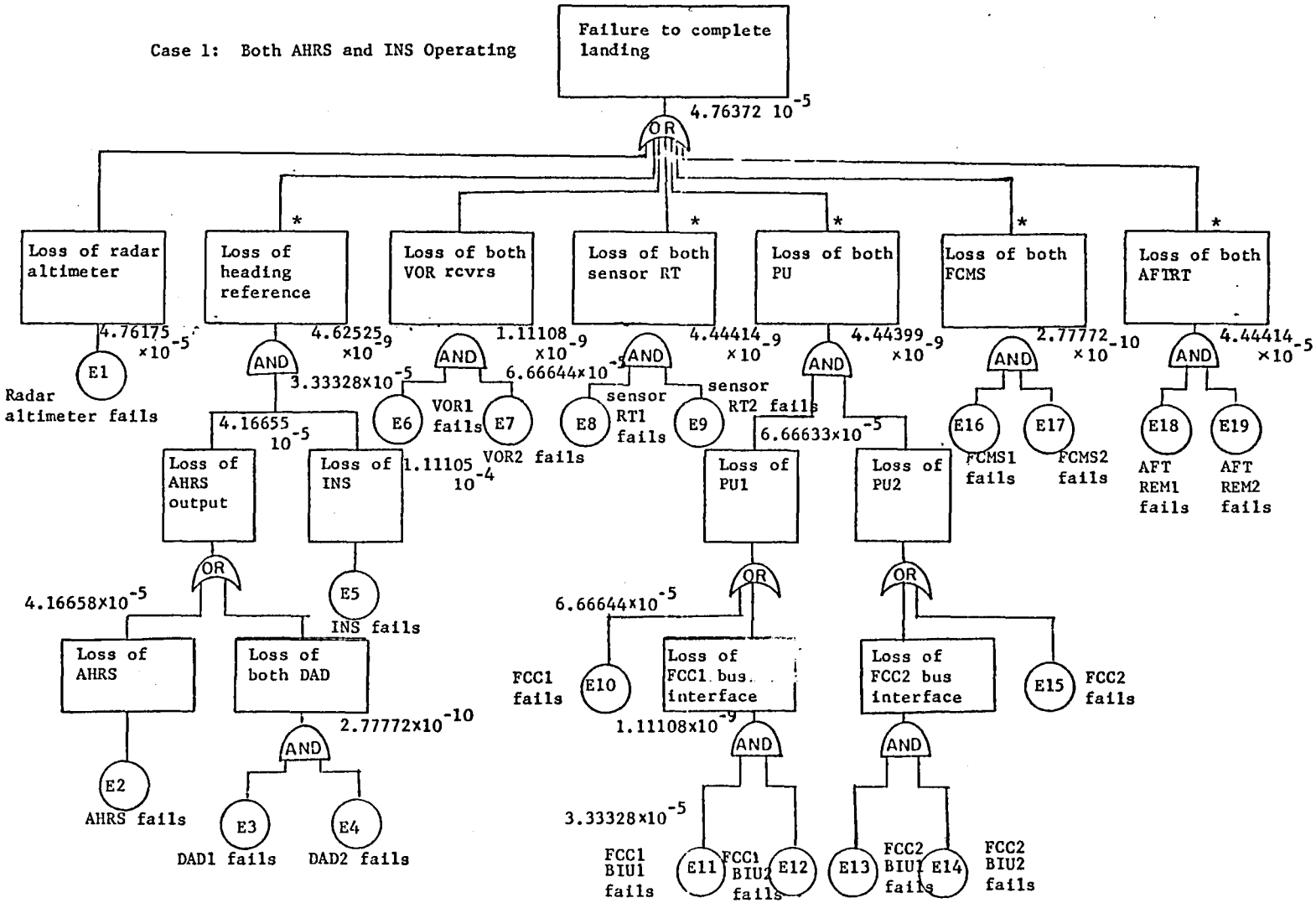


FIGURE 12. FAULT TREE FOR LANDING FAILURE, CASE 1

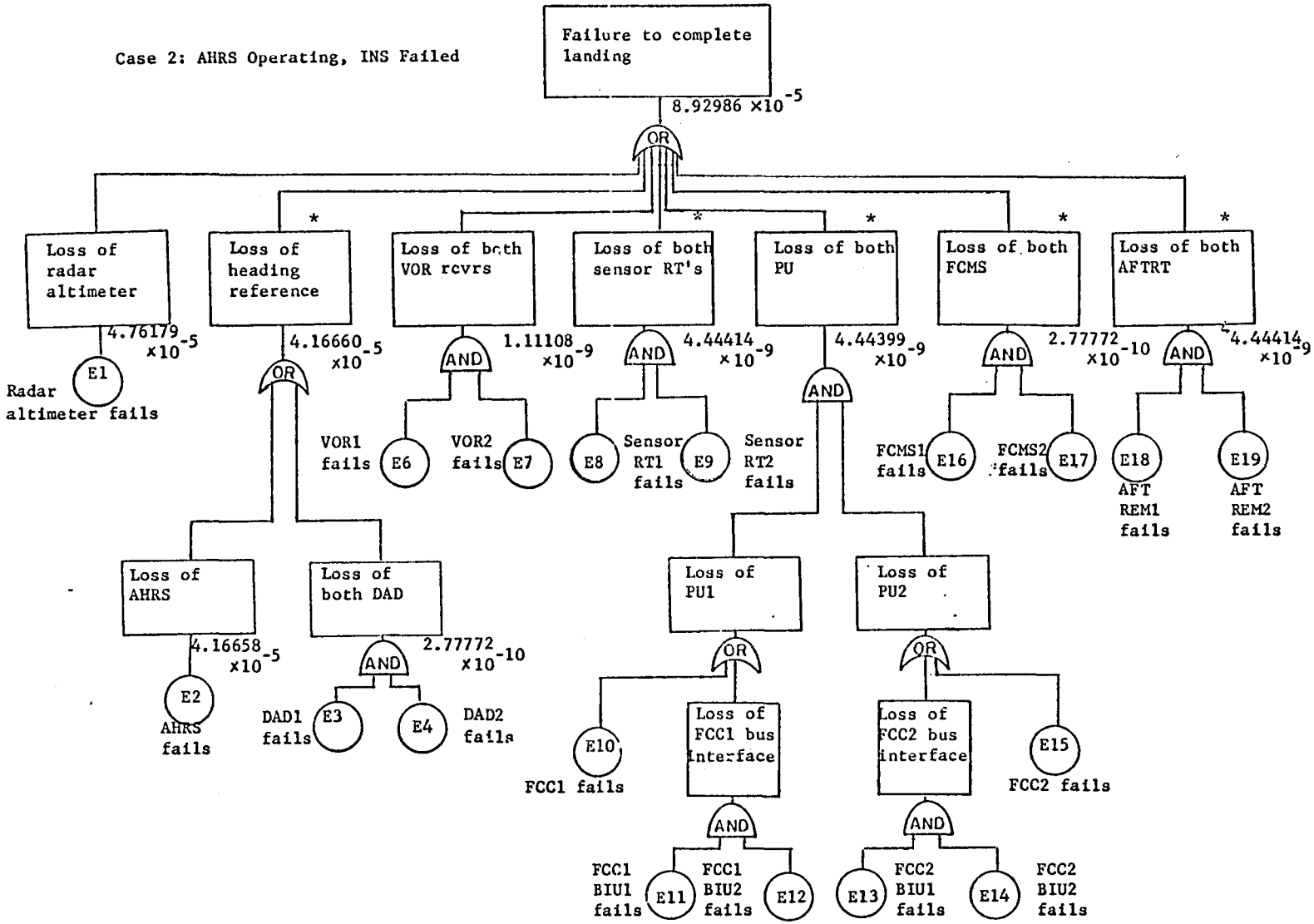


FIGURE 13. FAULT TREE FOR LANDING FAILURE, CASE 2

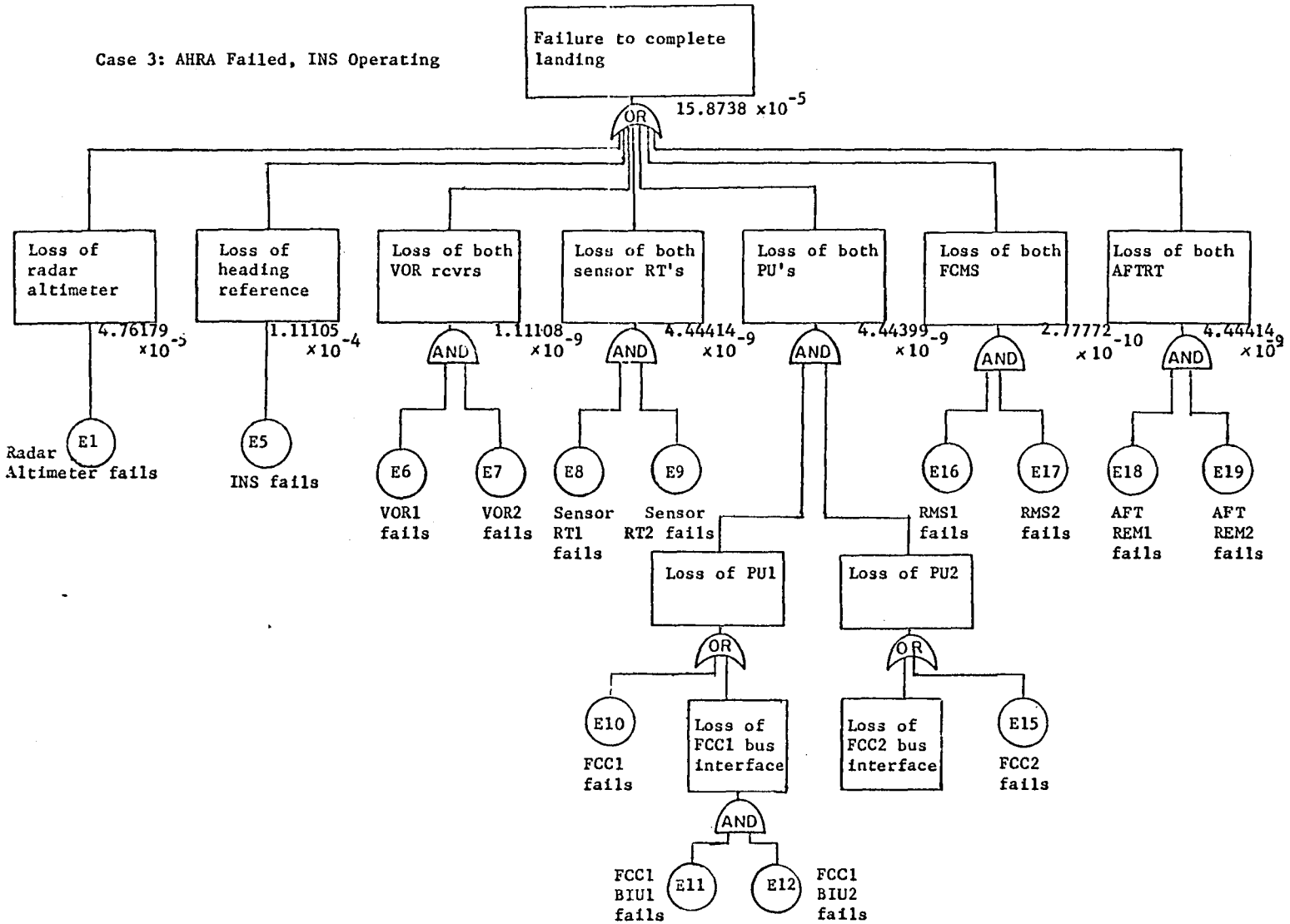


FIGURE 14. FAULT TREE FOR LANDING FAILURE, CASE 3

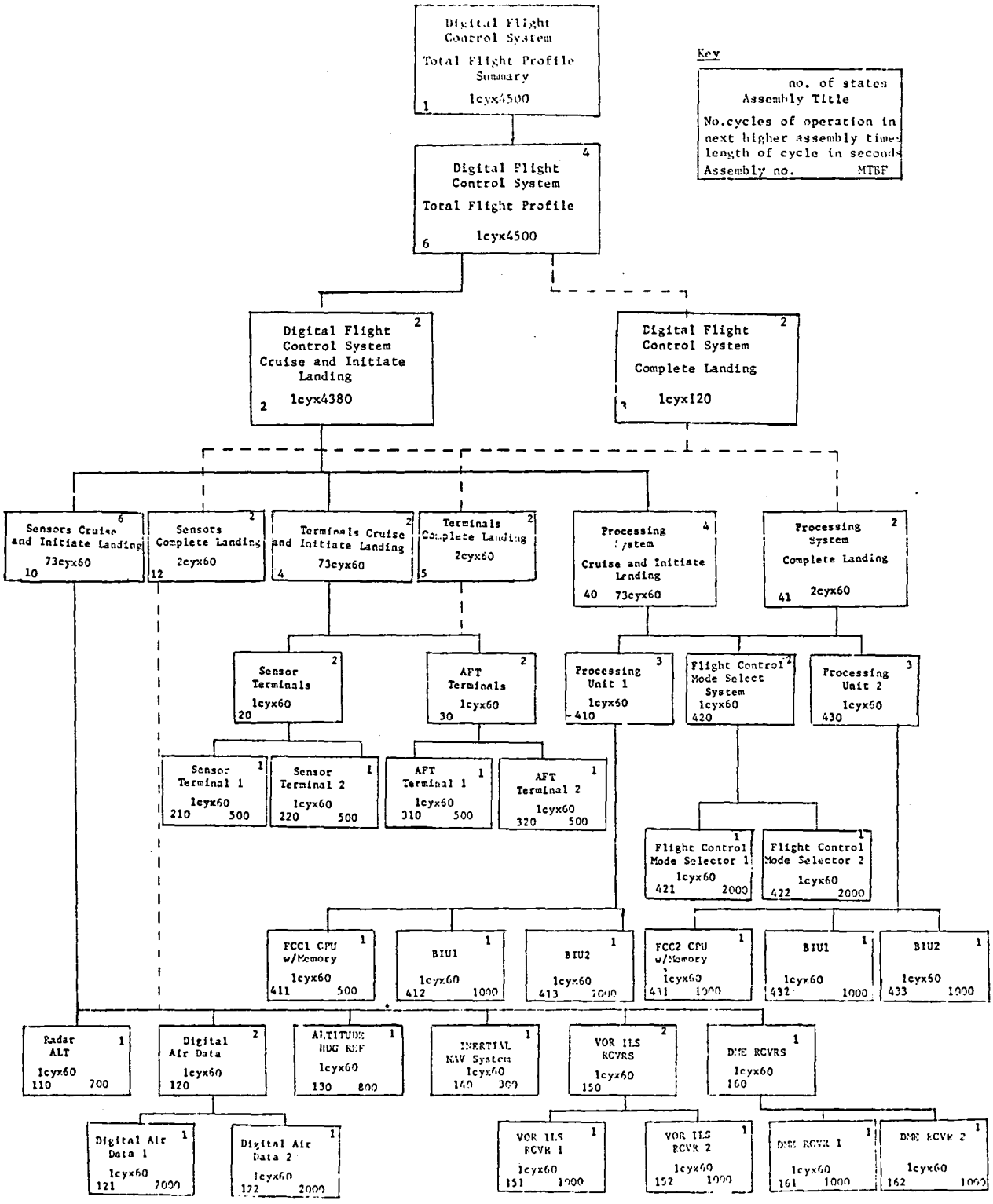


FIGURE 15. RELIABILITY BLOCK DIAGRAM

The second run corresponded to the landing phase. This breakdown was necessary to model the various states the system could occupy at the transition between the two phases. The results of the two runs were manually combined to derive the following results:

Pr (safe flight and Cat II landing) = 0.974236
Pr (safe flight and diversion) = 0.025740
Pr (loss of aircraft) = 23.69×10^{-6}

Solution Effort

The man-hours required to perform the TASRA analysis were estimated to be 25 hours. The actual time was somewhat greater, but it included effort spent resolving computer difficulties due to a system upgrade and interpreting the problem statement. In addition, 128 system seconds of computer time (on a CDC 6500) were used. This included creation and manipulation of input files and production of full TASRA documentation.

Discussion

TASRA was not designed to model multi-phase mission problems. It was therefore necessary to manually combine results for the different phases. This involved conditional probabilities. Some conceptual difficulty was encountered in ensuring the probabilities were correctly combined.

A considerable portion of the solution effort was devoted to the input logic tables. Every mathematically possible combination of subassembly states had to be evaluated for its effect on the assembly state. This task required detailed knowledge of the dual-dual system.

In addition to mission outcome probabilities, TASRA provided output on the unreliability "drivers". Also, additional computer runs to test variations of the system could be made using few man-hours.

Some of the numerical differences between TASRA and the other techniques can be attributed to the procedure used to combine subassembly probabilities into assembly probabilities. Each subassembly failure distribution is assumed to be exponential. TASRA assumes the assembly probabilities

are also from exponential distributions. The errors associated with this assumption are typically quite small. However, since the dual-dual problem involves small probabilities, the relative error may be significant.

MULTI-PROCESSOR SYSTEM ANALYSIS

Summary of Results

Table 12 summarizes the results of applying performability analysis and fault trees to the previously described multi-processor system. The performability analysis solution is described in the next subsection and is followed by the fault tree solution.

Performability Analysis of the Multi-Processor Problem

Analytic Summary

Five specific outcomes were required by the problem statement. These outcomes defined the accomplishment set:

$$A = \{a_0, a_1, a_2, a_3, a_4\}$$

where the mission outcome characteristics associated with each accomplishment level are:

a_0 - safe, on time, original destination;

a_1 - safe, late, original destination;

a_2 - safe, diverted to alternate destination;

a_3 - safe, aborted to point of origin;

a_4 - unsafe.

The mission (level 0) model used four binary variables to express the mission outcomes. The variables were as follows:

$$M_1 = \begin{cases} 0 \text{ if the flight is not aborted} \\ 1 \text{ otherwise} \end{cases}$$

TABLE 12. MULTI-PROCESSOR SYSTEM RESULTS FOR PERFORMABILITY ANALYSIS AND FAULT TREES

Quantities*	Performability	Fault Trees
P (safe, on-time, original destination)	0.99394882	0.99394863
P (safe, late, original destination)	0.00600770	0.00600766
E (penalty for late arrival)	$53.3502	$53.3525
P (safe, diversion)	3.5×10^{-9}	14.0×10^{-9}
E (penalty for diversion)	$00.0007	$00.0028
P (safe, aborted flight)	0	0
E (penalty for aborting)	0	0
P (aircraft lost, phase 1)	70.0×10^{-9}	69.99×10^{-9}
P (aircraft lost, phase 2)	869.0×10^{-9}	870.4×10^{-9}
P (aircraft lost, phase 3)	28.8080×10^{-6}	28.8957×10^{-6}
P (aircraft lost, phase 4)	10.3387×10^{-6}	10.3873×10^{-6}
P (aircraft lost, phase 5)	3.3978×10^{-6}	3.2921×10^{-6}
P (aircraft lost)	43.4835×10^{-6}	43.5155×10^{-6}
E (all penalties)	$53.3509	$53.3553
Man-hours for solution	59	22

* P indicates probability and E indicates expected value.

$$M_2 = \begin{cases} 0 \text{ if the flight is not diverted} \\ 1 \text{ otherwise} \end{cases}$$

$$M_3 = \begin{cases} 0 \text{ if the flight is on time} \\ 1 \text{ otherwise} \end{cases}$$

$$M_4 = \begin{cases} 0 \text{ if the flight is safe} \\ 1 \text{ otherwise.} \end{cases}$$

The level 0 trajectory space was the set of four dimensional vectors:

$$U^o = \left\{ \begin{bmatrix} M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix} \middle| M_i - 0, 1 \right\}$$

The subsets of U^o corresponding to each accomplishment level, denoted $\gamma_o^{-1}(a_i)$, were determined by inspection. They are shown in Table 13, where "*" represents "any possible value".

The function (level 1) model was based on characteristics of the multi-processor system and the specified criteria for aborting, diverting, late arrival, and safe flight. The criteria involved the number of fault-free processor triads and fault-free spare processors available during specific phases of the flight. A communication channel (i.e., an appropriate combination of sensor remote terminals, but interface units, and actuator remote terminals as specified in the problem statement) is required for safe flight. In addition, the existence of Category II weather was included since it impacts the need to divert. The function level variables were defined as follows:

F_{1J} = number of failed triads at the end of phase j

F_{2j} = number of spare processors at the end of phase j

$$F_{3j} = \begin{cases} 0 \text{ if a communication channel exists at the end of phase j} \\ 1 \text{ otherwise} \end{cases}$$

$$F_4 = \begin{cases} 0 \text{ if the weather at the original destination is not Category II} \\ 1 \text{ otherwise} \end{cases}.$$

The weather variable, F_4, is not phase dependent since weather information becomes known in phase 3 and does not subsequently change. In the matrices describing level 1 trajectories, the value of F_4 was indicated in Column 3.

TABLE 13. LEVEL 0 TRAJECTORY SETS FOR THE ACCOMPLISHMENT LEVELS

Level 0 Variables	Accomplishment Levels a_0	a_1	a_2	a_3	a_4
M_1	0	0	0	1	*
M_2	0	0	1	*	*
M_3	0	1	*	*	*
M_4	0	0	0	0	1

The other columns were filled with asterisks to indicate the lack of restrictions. The variables F_{1j} and F_{2j} are closely related since the ten processors are dynamically reconfigured to form as many fault-free triads as possible. Five phases were defined in the problem statement. The level 1 trajectory space was the following set of matrices:

$$U^1 = \left\{ \left[F_{ij} \right] \right\} \quad .$$

The next step was to determine the subsets of U^1 which corresponded to each accomplishment level. This was done by finding the matrices in U^1 corresponding to the level 0 trajectories for each accomplishment level. The following procedure was used. First, each M_i (level 0 variable) was considered individually. The level 1 matrices which result in a given value for each M_1 were determined. For example, consider $M_1 = 0$, which indicates "no abort". The flight is aborted if and only if one triad and no spares are available prior to the end of phase 2. The status of the communication channels and the weather have no bearing on the abort criteria. Table 14 shows the level 1 trajectories corresponding to $M_1 = 0$. Asterisks, which represent "any possible value", and entries such as "0 or 1" were used to reduce the number of matrices. Next, the level of trajectories were considered for each accomplishment level. For a given level 0 trajectory, the level 1 matrices for each M_i value were known. The corresponding level 1 matrices were constructed by forming all possible intersections using one matrix for each of the four M_i values. These sets of matrices were the level 1 inverses of the accomplishment levels and were denoted $\gamma_1^{-1}(a_i)$.

The base (level 2) model was defined in terms of the system components. One variable, N_j, was used to denote the number of processors which are failed by the end of phase j. N_j had integer values from zero to ten. A second variable, C_j, was set to zero if a communication channel exists at the end of phase j, and to one otherwise. Figure 16 displays the state diagram for the base model. Nine states of interest are identified. For convenience, the state numbers shown in the diagram were used to represent the state of the system. A base model trajectory was then represented as a vector of five state numbers, one for the end of each phase.

TABLE 14. LEVEL 1 TRAJECTORIES CORRESPONDING TO $M_1 = 0$ ("no abort")

$$\begin{bmatrix} 0 \text{ or } 1 & 0 \text{ or } 1 & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}$$

$$\begin{bmatrix} 0 \text{ or } 1 & 2 & * & * & * \\ * & 1 \text{ or } 2 & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}$$

$$\begin{bmatrix} 2 & 2 & * & * & * \\ 1 \text{ or } 2 & 1 \text{ or } 2 & * & * & * \\ * & * & * & * & * \\ * & * & * & * & * \end{bmatrix}$$

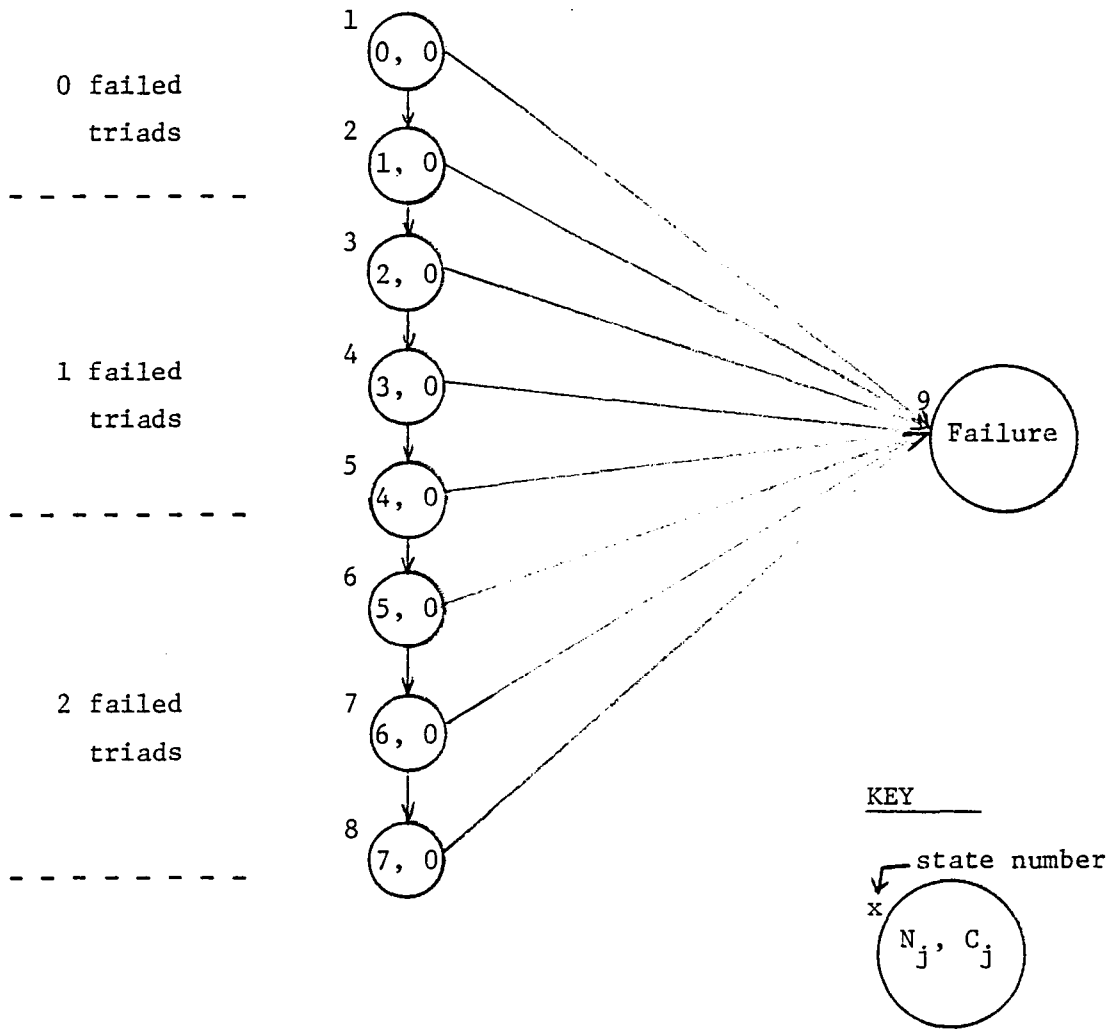


FIGURE 16. BASE MODEL STATE DIAGRAM

Base model trajectories for each level 1 trajectory were constructed directly from the $[F_{ij}]$ matrices. The first two rows of the matrices (number of failed triads and number of spare processors) correspond to the base model variable N_j. The third row of the matrix is equivalent to C_j. The fourth row only contained one variable (Category II weather), which was used in the probability computations. Grouping the base model trajectories corresponding to the level 1 trajectories for a given a_i (i.e., γ_1^{-1}, (a_i)) resulted in the set of base model trajectories for the outcome a_i, denoted $\gamma^{-1}(a_i)$. The method of construction caused the $\gamma^{-1}(a_i)$ trajectory sets to be Cartesian. They are summarized in Table 15.

The probability computations were performed using the basic probability equation shown in the synopsis of performability analysis. Table 16 displays the numerical results.

Solution Effort

A total of 59 man-hours were expended in the performability analysis solution of the multi-processor problem. The breakdown of man-hours by solution steps is:

Problem understanding, modeling development	18
Trajectory set computations	20
Probability computation	21
TOTAL	59

An additional 19 man-hours were expended on detailed computation checks to resolve differences with the fault-tree results. No significant errors were identified.

Discussion

Solution of the multi-processor problem using performability analysis required little ingenuity and substantial perseverance. It was, of course, necessary to have a good understanding of the multi-processor system and its mission requirements.

TABLE 15. BASE MODEL TRAJECTORY SETS FOR EACH ACCOMPLISHMENT LEVEL

Set Symbol	Set Trajectories
$\gamma^{-1}(a_0)$	{1, 2} x {1, 2} x {1, 2} x {1, 2} x {1, ..., 5}
	{1, 2} x {1, 2} x {1, 2} x {1, 2} x {6, 7, 8}
$\gamma^{-1}(a_1)$	{1, ..., 5} x {1, ..., 5} x {1, ..., 8} x {3, ..., 8} x {3, ..., 8}
	{1, ..., 5} x {1, ..., 5} x {1, ..., 5} x {3, 4, 5} x {3, 4, 5}
	{1, ..., 7} x {6, 7} x {6, 7, 8} x {6, 7, 8} x {6, 7, 8}
$\gamma^{-1}(a_2)$	{1, ..., 5} x {1, ..., 5} x {1, ..., 5} x {1, ..., 8} x {6, 7, 8}
	{1, ..., 5} x {1, ..., 5} x {6, 7, 8} x {6, 7, 8} x {6, 7, 8}
	{1, ..., 5} x {6, 7} x {6, 7, 8} x {6, 7, 8} x {6, 7, 8}
$\gamma^{-1}(a_3)$	{8} x S x S x S x {8}
	{1, ..., 7} x {8} x S x {8} x {8}
$\gamma^{-1}(a_4)$	{9} x Q x Q x Q x Q
	{1, ..., 8} x {9} x Q x Q x Q
	{1, ..., 8} x {1, ..., 8} x {9} x Q x Q
	{1, ..., 8} x {1, ..., 8} x {1, ..., 8} x {9} x Q
	{1, ..., 8} x {1, ..., 8} x {1, ..., 8} x {1, ..., 8} x {9}

where S indicates skipped phase

Q = {1, 2, 3, 4, 5, 6, 7, 8, 9}

TABLE 16. PROBABILITY (Pr) AND EXPECTED VALUE (E) RESULTS FOR PERFORMABILITY ANALYSIS OF THE MULTI-PROCESSOR PROBLEM

$P(a_0)=$	Pr (successful, on-time, original destination) = 0.993948817
$P(a_1)=$	Pr (successful, late, original destination) = 0.006007701
	E (economic penalty for late arrival) = \$53.3502
$P(a_2)=$	Pr (diversion, safe landing) = 3.5×10^{-9}
	E (economic penalty for diversion) = \$.0007
$P(a_3)=$	Pr (aborting, safe landing at origin) = 0
	E (economic penalty for aborting) = 0
	Pr (aircraft lost, phase 1) = 70.0×10^{-9}
	Pr (aircraft lost, phase 2) = 869.0×10^{-9}
	Pr (aircraft lost, phase 3) = 28.8080×10^{-6}
	Pr (aircraft lost, phase 4) = 10.3387×10^{-6}
	Pr (aircraft lost, phase 5) = 3.3978×10^{-6}
$P(a_4)=$	Pr (aircraft lost) = 3.4835×10^{-6}
	E (economic penalty) = \$53.3509

The models used in the model hierarchy were not difficult to define. The mission (level 0) and base (level 2) models were defined directly from the problem statement. Some latitude existed in selection of the function (level 1) model. In addition to the selected function model, options included using no function model, using only one phase (i.e., the entire mission), and treating the communication channels separately from the processors. The last option was based on the observation that the system could achieve any accomplishment level as long as a communication channel exists, while lack of a communication channel would result in loss of the aircraft. Separate treatment of the communication channels would have required less time but was not done in order to more accurately represent performability analysis.

Construction of the trajectory inverses from the mission model to the function model and then to the base model was conceptually straightforward but mechanically tedious. A simple procedure for naming the matrices (Reference 6) was useful for bookkeeping purposes. The time spent on the trajectories was divided about equally between computing the trajectories and checking the computations. At each model level, all mathematically possible trajectories were represented. A counting argument was used to check that the correct number of trajectories had been listed. Additional checks were made to ensure that no trajectories had been omitted or listed twice. These checks resulted in a high level of confidence that the base model accurately represented the problem.

The probability computations were conceptually easy, mechanical, and somewhat time consuming. Individual state transition probabilities were computed using the component failure rates and phase durations. The matrix multiplications consumed most of the time spent on probability computations. They could have been done in less time with METAPHOR (Reference 7), a computer program written for performability analysis computations. Also, METAPHOR would have significantly reduced the time spent checking the computations.

Fault Tree Method Solution

Analytic Summary

There are a number of different results required in this problem. Generally a different fault tree is required for each of the desired answers. There is some overlap in the computations required, but a distinct fault tree is necessary in each case.

Safe, On-Time Landing. The approach here will be to compute the complement of the desired probability (i.e., the probability of failure to arrive safely and on time. The fault tree with this as the top event is shown in Figure 17. The probabilities of the top event and the probabilities of the various contributing events are shown.

The probabilities of the individual events can be computed as follows. First, the probability of loss of flight management will be considered. This will be caused by a loss of two or more processors prior to the end of phase 4. The probability of this is

$$P_1 = 1-(1-p)^{10}-10p(1-p)^9 \qquad \text{(Eq. 1)}$$

where p is the probability of loss of a single processor in a period of 72 minutes. The probability p is given by

$$p = 1-\exp\{\frac{-72}{60x100}\}= 0.0119283 \qquad \text{(Eq. 2)}$$

Substituting this value in Equation 1 gives $P_1 = 0.0060078497$.

Next, it is necessary to establish P_2, the probability of loss of control during phase 5, given that flight management was intact at the end of phase 4. If, at the end of phase 4, 10 processors are operating, loss of control would require loss of five or more processors in three minutes. The probability of exactly five failures is $36p^5(1-p)^5$, where p is the probability of loss of a single processor in three minutes. This is of the order of 5×10^{-4}, which gives a probability of loss of the order of 10^{-17}, which is trivial compared to P_1.

It remains to determine P_3, the probability of loss of bus communication. This event is the top event of another fault tree, shown in Figure 18.

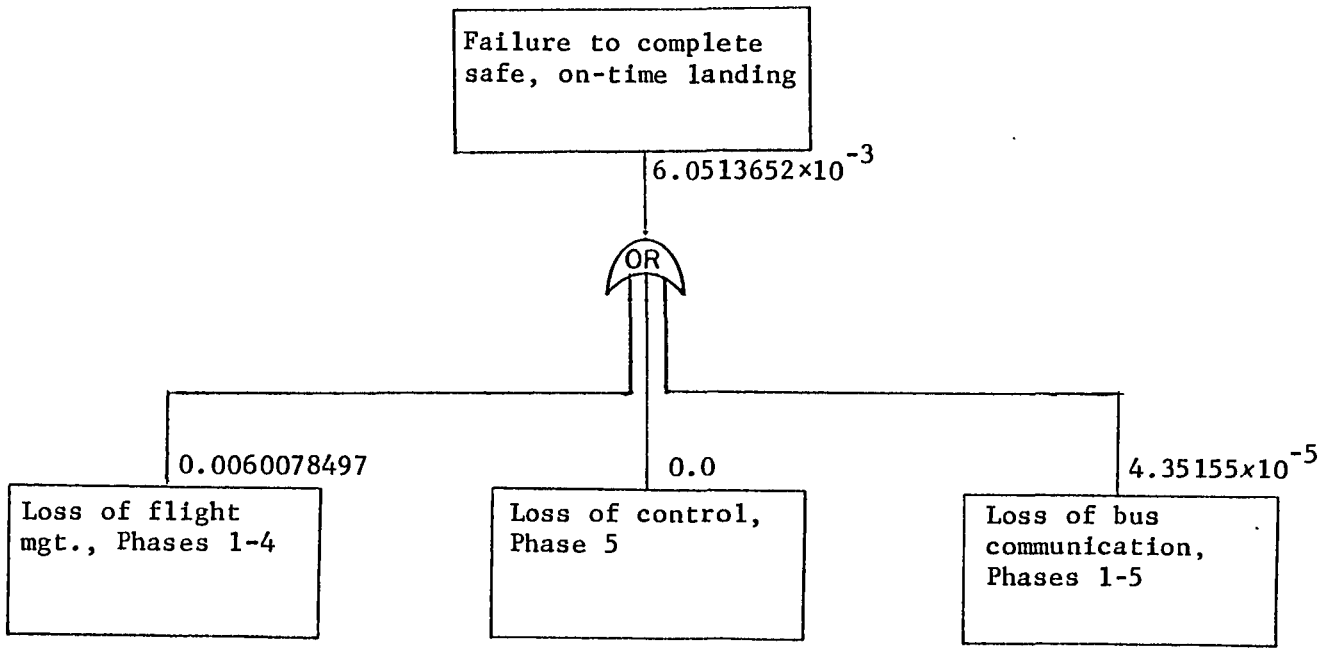
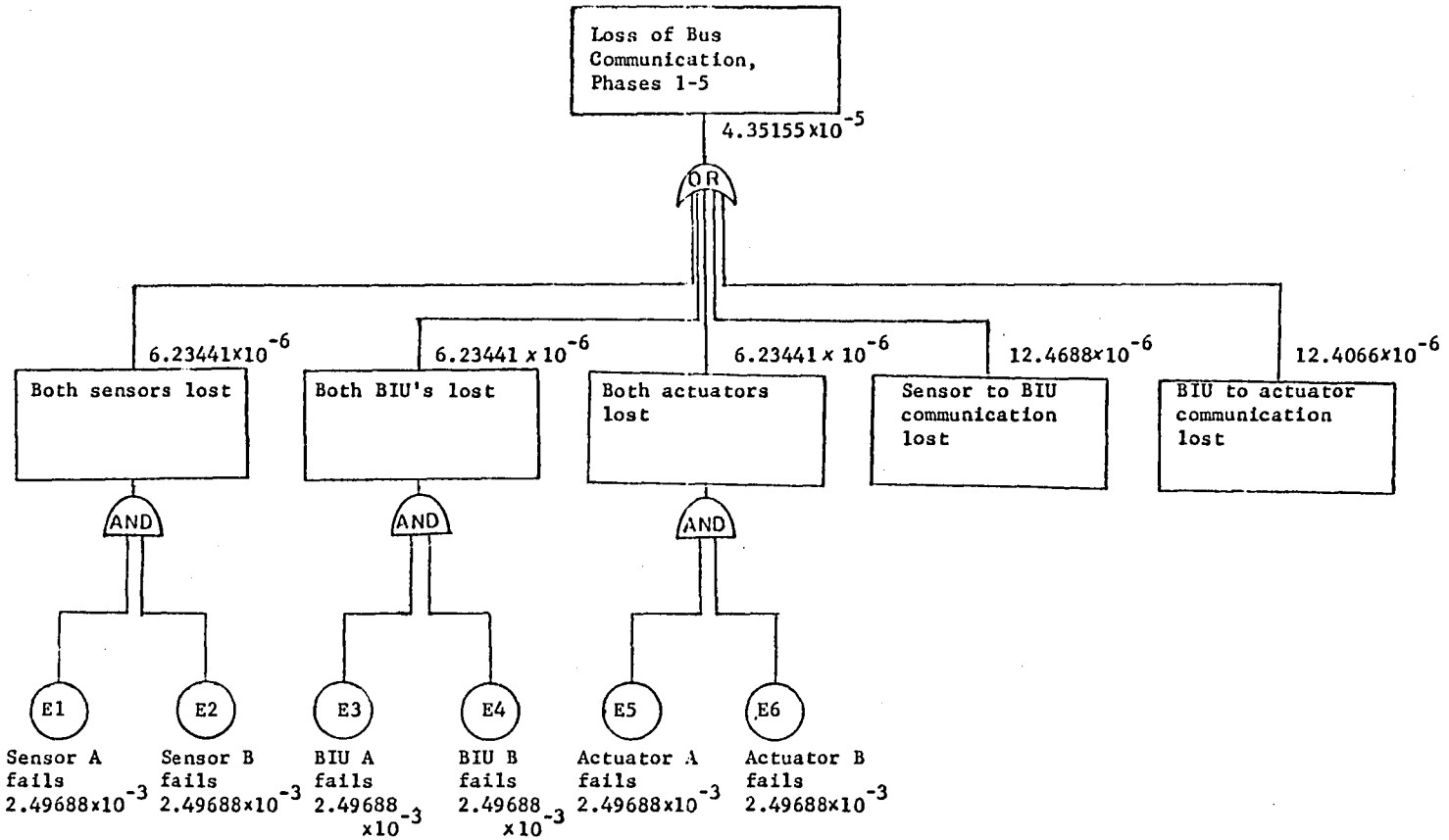


FIGURE 17. FAULT TREE FOR COMPLEMENT OF SAFE, ON-TIME LANDING



$$p=1 - \exp\left\{-\frac{75}{60 \times 500}\right\} = 2.49688\times10^{-3}$$

FIGURE 18. FAULT TREE FOR LOSS OF BUS COMMUNICATION

The first three contributing events can be treated in straightforward fashion. The loss of communication is somewhat more complex. Loss of Sensor to BIU communication can come about in two ways. Using Y to indicate fault-free operation and N to indicate failure, the two ways are depicted as follows:

Sensor A	Sensor B	BIU A	BIU B
Y	N	N	Y
N	Y	Y	N

The probabilities of these combinations of events can be computed from the fundamental failure probabilities.

Similarly, the loss of BIU to actuator communication can come about in two ways.

BIU A	BIU B	Actuator B	Actuator B
Y	N	N	Y
N	Y	Y	N

The total probability of loss of bus communication can then be determined by combining the failure probabilities of the individual contributing events.

Loss of Aircraft. This can come about in two ways which are indicated in the fault tree of Figure 19. The probability of loss of BIU communication in phases 1 through 5 has already been computed. The probability of loss of eight processors is given by p^8, where p is the probability of loss of a single processor in 75 minutes,

$$p = 1-\exp\{\frac{-75}{60x100}\}= 0.0119283 \quad .$$

p^8 is then of the order of 10^{-16}, which is trivial compared to the probability of loss of communication.

Successful, Late Landing at Original Destination. The fault trees for this case are shown in Figures 20 and 21. The situation is different depending on whether Cat II is required or not, so a separate fault tree must be prepared for each case.

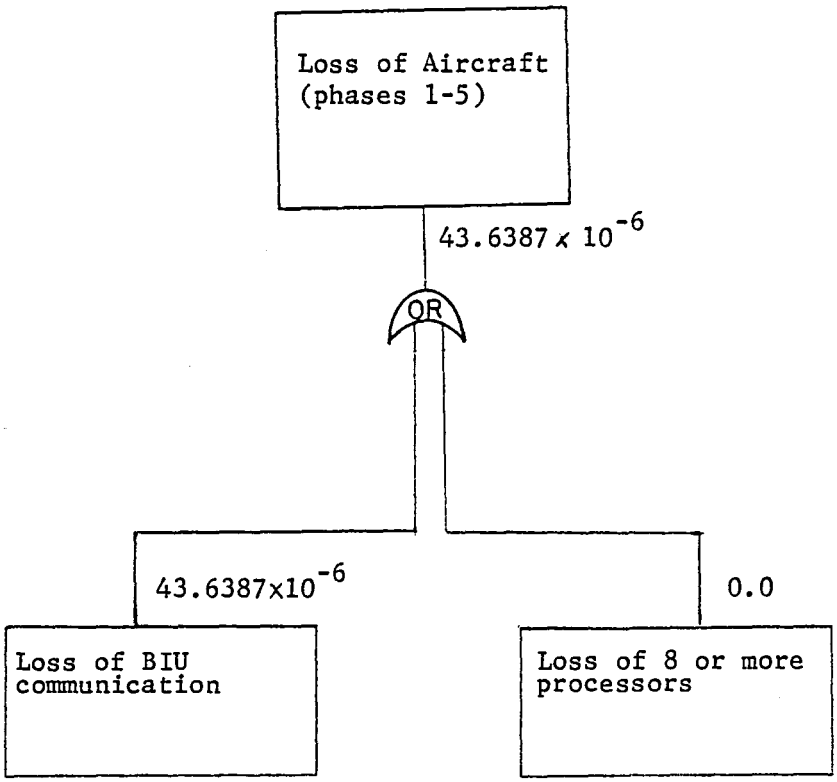


FIGURE 19. FAULT TREE FOR LOSS OF AIRCRAFT

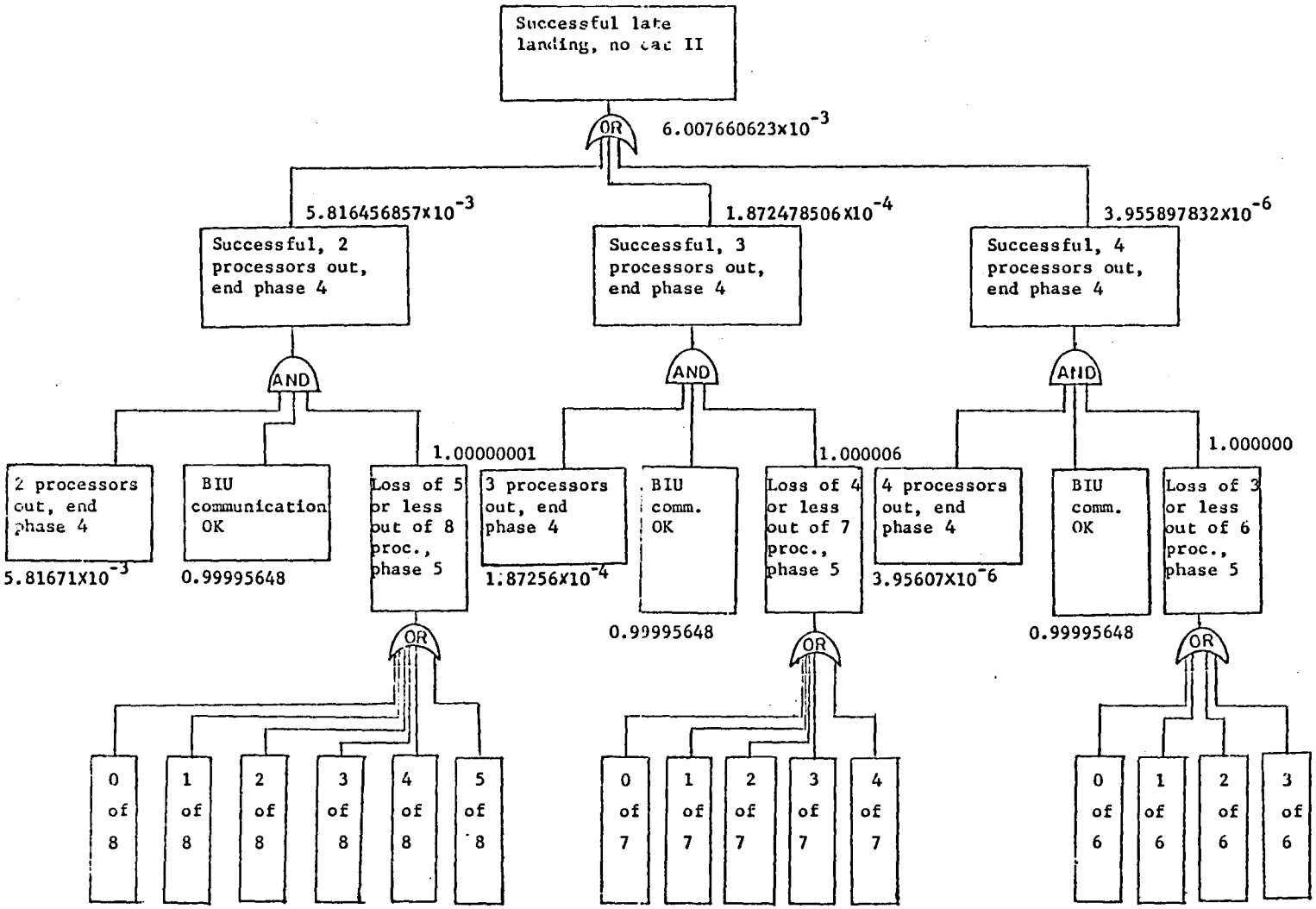


FIGURE 20. PROBABILITY OF SUCCESSFUL, LATE LANDING, NO CAT II REQUIREMENT

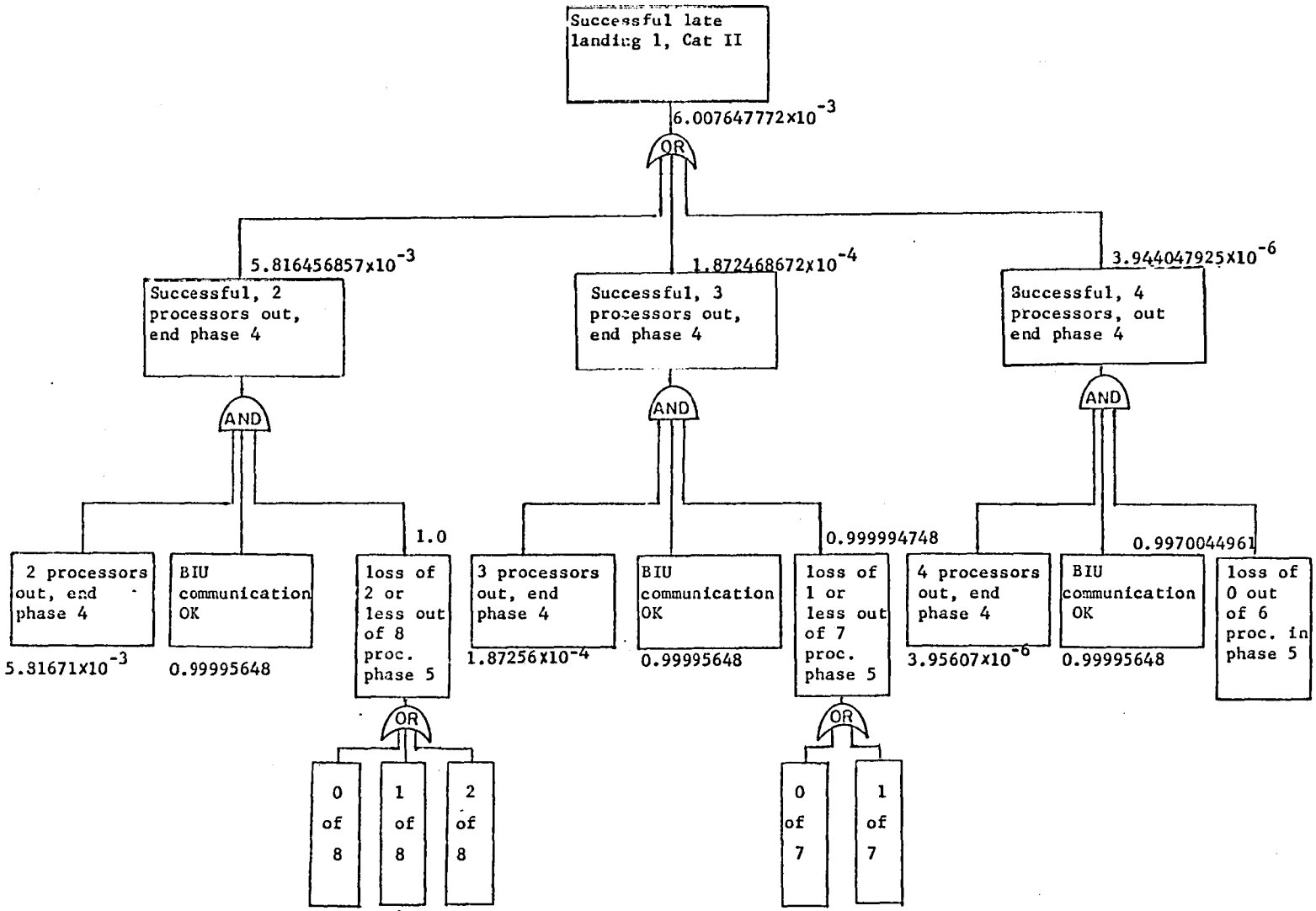


FIGURE 21. PROBABILITY OF SUCCESSFUL, LATE LANDING, CAT II REQUIREMENT

The total probability of a safe late landing is:

$$0.95 \times 6.007660623 \times 10^{-3} + 0.05 \times 6.007647772 \times 10^{-3}$$

$$= 6.00765998 1 \times 10^{-3} \quad .$$

Safe Diversion. To have diversion, it must happen that during phases 3, 4, or 5, Cat II capability is needed and not available. If Cat II capability is loss before 75 minutes, diversion will take place. This will happen if 5, 6, or 7 processors are lost and BIU communication is not lost and Cat II is required. The fault tree for this case is shown in Figure 22.

Abort. For this to occur in phase 1 it would be necessary to lose seven processors in three minutes. The probability of this is of the order 10^{-23}. The probability of aborting by the end of phase 2 is of the order 10^{-19}. Accordingly, the abort probability is taken as zero.

Collection of Results. In addition to the above computations, it was necessary to repeat some of the analyses on a phase-by-phase basis. These are done by repetition of the types of analysis given above. The results for this problem are summarized below.

Pr (successful, on-time landing, orig. destination)	=	0.993948634
Pr (successful late landing, orig. destination)	=	0.006007659
E (economic penalty for late arrival)	=	\$53.3525
Pr (diversion, safe landing)	=	14.04×10^{-9}
E (economic penalty for diversion)	=	\$0.002808
Pr (aborting, safe landing at origin)	=	0
E (economic penalty for aborting)	=	0
Pr (aircraft lost, phase 1)	=	69.993×10^{-9}
Pr (aircraft lost, phase 2)	=	870.38×10^{-9}
Pr (aircraft lost, phase 3)	=	28.8957×10^{-6}
Pr (aircraft lost, phase 4)	=	10.3873×10^{-6}
Pr (aircraft lost, phase 5)	=	3.2921×10^{-6}
Pr (aircraft lost)	=	43.5155×10^{-6}
E (economic penalty)	=	\$53.3553

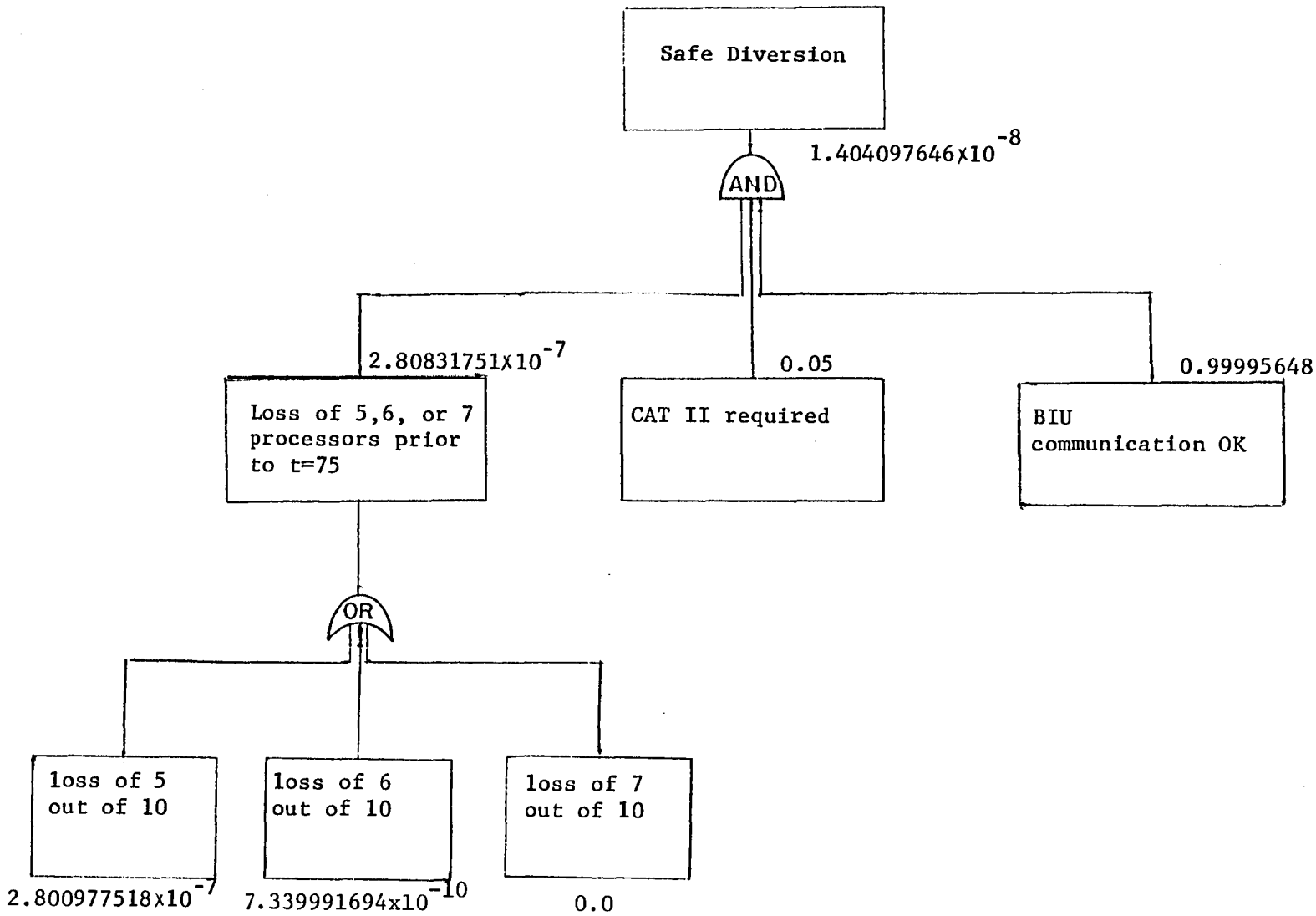


FIGURE 22. FAULT TREE FOR SAFE DIVERSION

Solution Effort

The time required for solution of this problem by the fault-tree technique was 22 man-hours. This included set-up, drawing of the fault trees, and all computations.

CROSS-TRAINING PROBLEM ANALYSIS

Summary of Results

Performability analysis and fault trees were applied to the cross-training problem, which was a simplified version of the multi-processor problem. The analyst who had been responsible for performability analysis applied fault trees to this problem. Performability analysis was applied by the analyst who had been responsible for fault trees.

Training requirements were quite different for the two techniques. Only one trainee man-hour was required for the fault-tree method. Twenty-eight trainee man-hours were expended learning performability analysis. In addition, approximately four hours of assistance from the performability analyst were used to clarify the written descriptions of the technique.

Fourteen man-hours were spent on the complete fault-tree solution. Performability analysis required 26 man-hours to establish the model hierarchy and compute the probability for one mission outcome. The complete computations are estimated to require 60 to 80 man-hours.

Performability Analysis Solution

Learning the Technique

In order to learn the method, two principal avenues were used. One was study of several papers and reports written by Meyer and his students, and the other was discussion with Michael Bridgman who did the major work on Meyer's method in the present study. It was found that the availability of Mr. Bridgman was a very great benefit in developing an understanding of the method. It would have taken several times longer without this resource.

The available papers are not designed for tutorial purposes, and they contain many points which are difficult to understand at first reading. If the technique is to become widely used, it may be necessary to develop materials which are (1) more comprehensible and (2) contain better motivation for the reader in terms of explaining the advantages of Meyer's method over other methods.

Analytical Summary

Problem Structure. There is more than one way to fit the problem into Meyer's format. The one selected here seems to be logical, but there are certainly others which could be defended. Three levels were defined: (1) the accomplishment set, (2) the "aircraft level", and (3) the base process. The accomplishment set is defined as follows:

a_o = successful, on-time landing at original destination

a_1 = successful but late landing at original destination

a_2 = safe diversion

a_3 = safe abort

a_4 = loss of aircraft

The overall objective is to determine the probabilities of these various outcomes.

The aircraft level is concerned with the capability of the avionics system during each phase of the flight. A trajectory at the aircraft level is defined by a vector,

$$[q_1 \quad q_2 \quad q_3 \quad q_4 \quad q_5 \quad x_6]$$

where

$$q_i = \begin{cases} 3 \text{ if there is full capability at the end of phase i} \\ 2 \text{ if only augmentation and control are operating at the end of phase i} \\ 1 \text{ if only augmentation is operable at the end of phase i} \\ 0 \text{ if all capability is lost at the end of phase i} \end{cases}$$

$$x_6 = \begin{cases} 1 \text{ if Cat II capability is required} \\ 0 \text{ if Cat II capability is not required} \end{cases}$$

The base level trajectory is defined by the vector

$$[x_1 \quad x_2 \quad x_3 \quad x_4 \quad x_5 \quad x_6]$$

where x_i, i = 1, 2, 3, 4, 5, is the number of LRU's operating at the end of phase i and x_6 is as defined above.

The three levels, and the mapping between them, can be portrayed as follows:

Accomplishment Set
K_o
Aircraft Level
K_1
Base Model

Mapping From Aircraft Level to Accomplishment Set. Rather than define the complete mapping K_o, it was decided to consider only the accomplishment level a_o. In this problem, the aircraft level trajectories which produce a_o can be enumerated. They are:

$$\begin{array}{lllllll} V_1: & (3 & 3 & 3 & 3 & 3 & *) \\ V_2: & (3 & 3 & 3 & 3 & 2 & *) \\ V_3: & (3 & 3 & 3 & 3 & 1 & 0 \end{array} \qquad (1)$$

where, following Meyer's notation, the symbol * is used to denote a case in which the component can take on any value on its range.

Mapping From Base Model to Aircraft Level. It remains to establish the mapping from the base model to these aircraft level trajectories. From the definition of the problem, it can be seen that the first of the trajectories (1) is produced by the Cartesian trajectory set:

$$V_1 = \{9,10\}x\{9,10\}x\{9,10\}x\{9,10\}x\{9,10\}x\{*\} \qquad (2)$$

Computation of Probability. To evaluate the probability of a trajectory set of this type, the following result of Wu and Meyer (Reference 9, Theorem 1) may be used:

$$Pr(V) = I(0)\left[\prod_{k=1}^{5} P_k G_k\right] F \tag{3}$$

where I(0) is a row vector of the probabilities of being in the various states x = 0, 1, 2, ..., 10 at the start of the problem, F is a column vector all of whose elements are unity. P_k is a state transition matrix whose elements are $P_k(i,j)$ = probability of being in state j at the end of phase k, given that system was in state i at the beginning of phase k. G_k is a matrix defined by

$$g_k(i,j) = \begin{cases} 1 \text{ if } i = j \text{ and } i \varepsilon R_k \\ 0 \text{ otherwise} \end{cases}$$

where R_k is the set of allowed states in V during the k_{th} phase.

Evaluation of Equation 3 is rather complex. It involves multiplication of ten matrices. In this case, the problem is somewhat simplified by that fact that most of the components are zero, and the matrices to be multiplied are actually only 2 x 2. This is still a substantial computation task, however. For each phase, the product $P_k G_k$ is of the form

$$\begin{bmatrix} p(0,0) & \cdots & p(0,10) \\ p(1,0) & & \\ \vdots & & \\ p(10,0) & \cdots & p(10,10) \end{bmatrix} \begin{bmatrix} & 0 & & 0 \\ & & & \\ & & & \\ 0 & & 1 & 0 \\ & & 0 & 1 \end{bmatrix} = \begin{bmatrix} & 0 & & 0 \\ & & & \\ & & & \\ 0 & & p(9,9) & p(9,10) \\ & & p(10,9) & p(10,9) \end{bmatrix} \tag{4}$$

Since $P_k(9,10) = 0$ for all k, it is necessary only to compute the three remaining probabilities.

$$\begin{aligned} P_k(9,9) &= (1-U_k)^9 \\ P_k(10,9) &= 10U_k(1-U_k)^9 \\ P_k(10,10) &= (1-U_1)^{10} \end{aligned} \tag{5}$$

where U_k is the probability of failure of a single LRU during phase k.

$$U_1 = 1-\exp\{\frac{-3}{60x100}\} = 4.9987510x10^{-4}$$

$$U_2 = 1-\exp\{\frac{-8}{60x100}\} = 1.3324449x10^{-3}$$

$$U_3 = 1-\exp\{\frac{-51}{60x100}\} = 8.4639771x10^{-3} \qquad (6)$$

$$U_4 = 1-\exp\{\frac{-10}{60x100}\} = 1.6652786x10^{-3}$$

$$U_5 = U_1$$

Substituting the values of (6) into the transition probabilities of (5), and substituting those results into the matrices indicated in (4), and multiplying the five resulting matrices in the proper order gives

$$P_r(V_1) = \begin{bmatrix}0 & 1\end{bmatrix} \begin{bmatrix}0.89359715 & 0 \\ 0.11100440 & 0.88249671\end{bmatrix} \begin{bmatrix}1 \\ 1\end{bmatrix} = 0.99350111 \qquad (7)$$

The second vector in (1) implies the Cartesian set

$$V_2 = \{10,9\}x\{10,9\}x\{10,9\}x\{10,9\}x\{8,7,6\}x\{*\} \qquad (8)$$

while the third vector in (1) implies

$$V_3 = \{10,9\}x\{10,9\}x\{10,9\}x\{10,9\}x\{3,4,5\}x\{0\} \qquad (9)$$

All these trajectory sets are identical through the first four phases. Making the necessary changes in the fifth phase, the resulting probabilities are:

$$P_r[V_2] = 4.906859x10^{-4}$$
$$P_r[V_3] = 0(10^{-12}) \qquad (10)$$

The probability of a_o is, then, the sum of the three probabilities.

$$P_r[a_o] = 0.99399179 \qquad (11)$$

Solution Effort

A total of 28 man-hours were expended learning the method to the degree needed to solve the sample problem. An additional 26 hours were required to compute the results given above. The majority of this computation time was used in multiplying the matrices.

It should be kept in mind that only the accomplishment level a_o was considered. A complete solution of the problem would have required evaluation of the other four accomplishment levels. Each would require a time approximately the same as that expended here. All computations were done by hand using a desk calculator.

Obviously, computer assistance would greatly reduce the time required and the total resources required for a solution. For this problem, however, it seems clear that Meyer's method requires a computation time perhaps an order of magnitude greater than that for the corresponding fault-tree solution.

Fault Tree Solution of Cross-Training Problem

Analytic Summary

Five mission outcomes were specified by the problem statement. One fault tree was constructed for each outcome. The fundamental events for a given tree specified the number of processors which were failed at the ends of particular phases and the presence or absence of Category II weather. In some cases involving "AND" logic gates, one event was conditioned upon occurrence of another event. The probability equations were written directly using the fundamental events for each tree. The numerical results are shown in Table 17.

Solution Effort

Constructing the fault trees and computing the probabilities required ten man-hours. Since the sum of the probabilities of all outcomes did not equal 1.0, an error was indicated. Two man-hours were expended finding the error (which was a multiplication error). Two more man-hours were spent

TABLE 17. PROBABILITY RESULTS OF FAULT TREE ANALYSIS OF CROSS-TRAINING PROBLEM

Mission Outcome	Probability
Safe, on-time, original destination	0.993992
Safe, late, original destination	0.006008
Safe, diversion	4×10^{-9}
Safe, aborted (land at origin)	0
Loss of aircraft	2×10^{-14}

checking other computations. The total time expended on the problem was 14 man-hours.

Discussion

The fault trees for the cross-training problem were directly constructed from the problem statement. No significant difficulties were encountered. Some assistance was gained from having solved the multiprocessor problem, which was simplified to create the cross-training problem.

Conditional combinations of fundamental events were used to express outcomes sensitive to the phase in which a certain level of degradation is realized. Care was required to ensure that all combinations resulting in each outcome were included. In addition, care was required in writing the correct probability expressions for the fundamental events.

CONCLUSIONS AND RECOMMENDATIONS

The objective of this study was to assess performability analysis in terms of its capabilities, practical usefulness, and costs of application. The assessment method was to solve sample problems using performability analysis and fault trees and then compare results. One analyst was assigned to each technique. The assignment was reversed for the last sample problem. The analysts had neither learned nor applied either technique prior to this study. An automated technique, TASRA, was used in two problems for further comparison.

Preceding sections of this report synopsized the techniques, presented the sample problems, and summarized the results of analyzing each problem with the various techniques. This section discusses the conclusions and recommendations derived during this investigation.

LEARNING REQUIREMENTS

Much more time and effort is required to learn performability analysis than to learn the fault tree approach. "Learn" is assumed to include "understand the underlying theory". Although formal material could be helpful for learning fault trees, it is not required. The basic fault tree approach is conceptually simple and can be learned in a matter of hours. Performability analysis could require a man-week or more to attain the same level of understanding using currently available material. The concept of functional dependencies, the model hierarchy, and the computational methods all contribute significantly to the requirements. Tutorial material, which does not currently exist for performability analysis, could reduce the learning time. Even with such material, performability analysis will still require more time and effort to learn.

The analysts assigned to performability analysis and fault trees had solid mathematical backgrounds. They both found that performability analysis required much more mathematical background than fault trees. The nature of the relationship between level of mathematical background and the time and effort required to learn each technique is diagrammed in Figure 23. The vertical

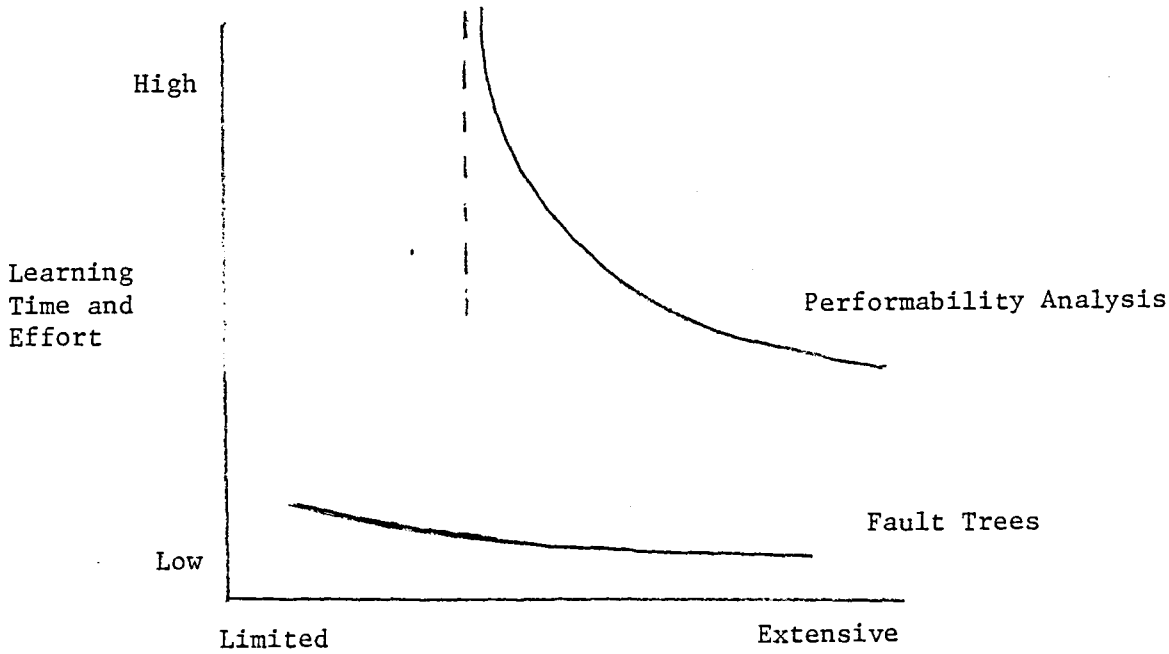


FIGURE 23. CONCEPTUAL RELATIONSHIP BETWEEN LEARNING REQUIREMENTS AND MATHEMATICAL BACKGROUND

asymptote for performability analysis indicates that minimum background requirements for understanding the technique are much greater than fault trees. Concepts related to the asymptote are composition of functions, inverse functions, projection mappings, set manipulations, and matrix multiplication.

APPLICATION EFFORT

For the sample problems* used in this investigation, performability analysis required significantly more solution effort than the fault tree approach. For the dual-dual problem, TASRA required less effort than fault trees or performability analysis, but TASRA was not exercised to the same level of conceptual accuracy. Table 18 summarizes the man-hours expended for the different problems. The figures shown include time to become familiar with the problem as well as modeling and computation times.

The dual-dual problem was found to have characteristics beyond the designed capabilities of TASRA. In particular, TASRA was not structured to handle multiple-phase missions involving dependencies among functions or components. Some hand manipulations were necessary to approximate the logical connections between phases. Since no significant information was gained by applying TASRA to the dual-dual problem, it was not applied to the other two problems. TASRA is not discussed any further in this report.

The time differences between performability analysis and the fault tree approach are believed to represent differences between the two techniques and not differences in analyst capability. The sample problems did not involve details of flight control or computing systems which could give one analyst an advantage regardless of solution technique. In addition, the cross-training problem solutions exhibited solution time differences similar to those of the dual-dual and multi-processor problems.

Performability analysis utilizes a hierarchy of models to connect the mission outcomes of interest (i.e., the accomplishment levels) to sets of possible component behaviors. The model hierarchies were not uniquely defined by the problem statements. The time required to define and select a hierarchy

---

* The series-parallel problem is excluded from this discussion because its extreme simplicity provides a poor basis for comparing techniques.

TABLE 18. SOLUTION MAN-HOURS SUMMARY

	Man-Hours for Solution*		
	Performability	Fault Trees	TASRA
Dual-Dual	46	30	25
Multi-Processor	59	22	--
Cross-Training	26**	14	--

* Includes model construction and computations. Does not include detailed computational checks.

** Represents partial problem solution as described in the text.

was approximately twenty percent of the total solution time for the dual-dual and multi-processor problems.

The process of determining the set of base model trajectories associated with each mission outcome consumed about one third of the total solution time. Every mathematically possible trajectory is expressed at each step in the model hierarchy, even if it has a zero probability of occurrence or is not physically possible. This is inefficient in terms of time requirements. However, as noted under the heading "Solution Accuracy", this allows for a logical correctness test which can increase confidence in the accuracy of the solution.

Probability computations accounted for 40 to 50 percent of the solution time for performability analysis. A state transition matrix was required for each mission phase. All possible transition probabilities had to be computed. Matrix multiplications were then performed for each mission outcome. Many quantities were zero or negligible (less than 10^{-12}), but time was still spent on them. A significant amount of time was spent checking the computations for numerical accuracy.

The fault-tree approach focused only on events of interest. Each mission outcome required a separate fault tree. Physically impossible events or combinations of events were not included in either the trees or the associated probability computations.

Expressing dependencies among functions or components, for one or several phases, in terms of fault trees, required some ingenuity and the use of conditional probabilities and combinations of events. Performability analysis has a structure oriented towards capturing dependencies. Dependencies are expressed at the model level (e.g., mission, function, component) at which they occur. The procedure for determining the base model trajectories associated with each mission outcome maintains all dependencies expressed at intermediate model levels.

The sample problems involved a small number of functional dependencies. Examples include the related and interdependent requirements for the digital air data, AHRS, and INS in the dual-dual problem and the conditions for diversion in the multi-processor problem. The presence of few dependencies is viewed as an advantage for fault trees.

A few dependencies, multiple phases, and several mission outcomes characterized the sample problems. For each outcome, the fault-tree approach only considered combinations of events resulting in that outcome. Performability analysis, on the other hand, considered all possible combinations of events.

SOLUTION ACCURACY

Performability analysis was found to have no inherent characteristics which make it more or less numerically accurate than other techniques. However, for very complex problems, performability analysis may result in a higher level of confidence that no mistakes have been made. Many of the set manipulations for determining base model trajectories are mechanical in nature and can be readily checked. At each level in the model hierarchy, counting procedures can be used to ensure the correct number of trajectories have been expressed. The actual probability computations involve matrix multiplications which are tedious but can be checked. Also, the matrix computations have been automated (Reference 4). Fault-tree analysis, on the other hand, can involve conditional probabilities and clever modeling, both of which are more difficult to verify.

SUMMARY OF CONCLUSIONS

Conclusions based on this investigation can be summarized as follows:

- It is possible to learn and apply performability analysis using existing descriptive material.
- Performability analysis requires much more effort to learn and understand than fault trees.
- For the sample problems, performability analysis required more effort than fault trees.

RECOMMENDATIONS

Implications of Complex Problems

As noted above, performability analysis required more solution effort than the fault-tree method for the sample problems. The dual-dual and multiprocessor problems were only moderately complex. More complex problems can easily be envisioned. This recommendation is concerned with the effects of applying the two techniques to more complex problems.

Figure 24 diagrams the hypothesized conceptual relationship between problem complexity and solution effort for fault trees and performability analysis. Complexity can be described in terms of the numbers of outcomes of interest, dependencies, and mission phases, and of the fault types to be considered. Solid lines are used in the region of the graph represented by the sample problems. Dashed lines represent hypothesized behavior of the techniques.

The hypothesized behavior is based on the following factors:

- Familiarity with performability analysis
- Study of application of performability analysis to the SIFT computer (Reference 3)
- Extrapolations based on the sample problems.

Several technique characteristics which support the hypotheses are described in the following paragraphs.

Consider the fault-tree approach. Each mission outcome requires a separate fault tree. Increasing the number of phases tends to increase the number of fundamental events which must be considered. Increasing dependencies could cause the solution requirements to increase dramatically in terms of time and ingenuity because of logical interconnections among dependencies and a large number of possible event combinations.

Performability analysis requirements appear to be less sensitive to increases in outcomes, phases, and dependencies. All base model trajectories are included regardless of the number of dependencies. More outcomes simply require the trajectories to be divided into more sets. More matrix multiplications are also required. Additional phases tend to increase the number of mechanical steps but do not require a great deal in terms of analyst ingenuity.

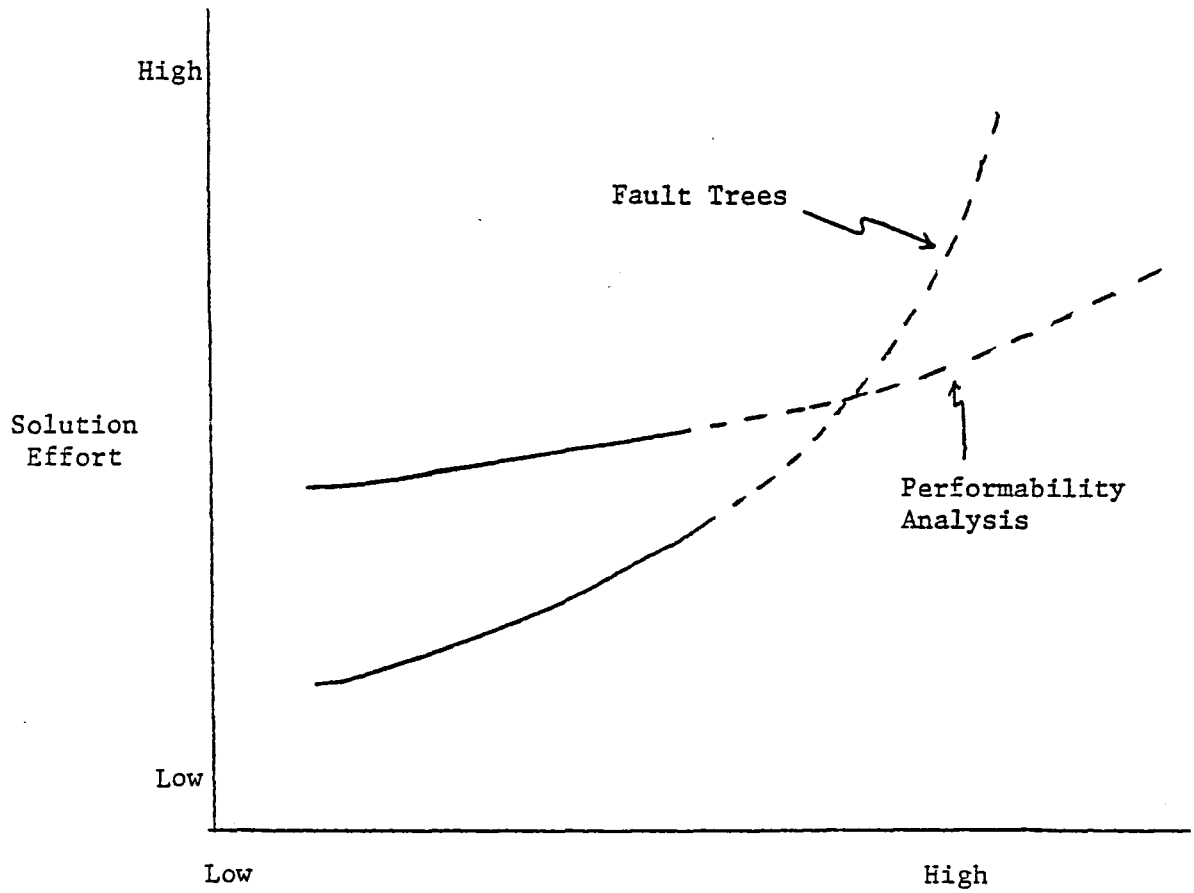


FIGURE 24. HYPOTHESIZED CONCEPTUAL RELATIONSHIPS BETWEEN COMPLEXITY AND SOLUTION EFFORT

Only permanent faults were treated in the sample problems. Transient faults are quite difficult to handle with fault trees. The only known approach is to treat components subject to transient faults with availability equations, which tend to become complex. A component subject to transient and permanent faults may need to be treated as two components. This could cause fault trees to become very cumbersome. Performability analysis uses a state approach, which lends itself more readily to modeling transient faults (this issue is addressed in more detail below).

It is recommended that a highly complex problem be investigated using performability analysis and fault trees to determine if the relationship depicted in Figure 24 is conceptually accurate. One approach would be to analyze the SIFT computer problem (Reference 3) and then compare results and effort with the performability solution. A second approach would be to define a new problem such as a next-generation transport aircraft (post-Boeing 767) with computers based on the FTMP architecture (Reference 15), and apply both techniques to the problem.

Ability to Model Transient Faults

Faults may be classified as either permanent or transient. The sample problems only considered permanent faults. The ability of performability analysis to model transient faults was not addressed by the sample problem solutions. However, study of the technique indicated that it could handle transient faults through appropriate definition of the base model. This claim could be verified by defining a problem involving transient faults and then proceeding with performability analysis until solution feasibility is clearly established.

Software Errors

It is recommended that no attempt to include software error models in fault trees or performability analysis be made at this time. No validated fundamental model of software errors is known to exist. Consequently, it it not feasible to determine if one technique is preferable in terms of modeling software errors.

Tutorial Material for Performability Analysis

The material used for learning performability analysis consisted of status reports by Meyer (References 1, 2, and 3) and technical papers (References 5-9). The status reports focus on technical developments achieved during the reporting period. The technical papers focus on particular aspects of the technique. None of this material was written for tutorial purposes. As a result, the effort to learn performability analysis was much greater than necessary. Tutorial material to explain the theory and application of the technique should be developed.

Performability Analysis Tools

A large proportion of the effort in applying performability analysis is mechanical in nature. Automated tools could potentially reduce the time and effort required to derive solutions. An interactive computer program for the probability computations exists (Reference 4). It should be validated.

Two potential areas for tools are model building and formulation of the capability function (including computation of the base model trajectory sets corresponding to the mission outcomes). A tool for the first area would probably be an interactive aid. The second area might be amenable to complete automation. It is recommended that these possibilities be investigated.

OBSERVATIONS

This section presents several observations on the evaluation of fault-tolerant computing systems. They are neither conclusions nor recommendations, but they reflect important practical considerations which came to light during the study.

Credibility of Solution

Reliability of a fault-tolerant computing system is a complicated and sensitive exercise. Such systems have complex structures and logic paths. The desired system failure probability is typically so small that the numerical

techniques used in the analysis may cause significant errors. Currently available reliability models have limitations in these areas of adaptability to system configurations and numerical accuracy. More capable models are also more difficult to exercise. Again, since the numbers of interest are so small, a slight computational or procedural anomaly could cause significant error. It may therefore be difficult to produce a solution which is uniformly accepted as credible.

One approach to enhancing solution credibility would be to apply more than one technique to the system reliability problem. They could be applied by the same or different personnel. The goal would be to obtain concurrence on the result.

Data Support of Models

Accurate reliability estimation of ultra-reliable systems requires two key ingredients: a model with sufficient fidelity and data to support that model. The models applied in this study do not precisely capture all system characteristics (e.g., recovery strategies, timing difficulties), but it appears they can provide much more modeling precision than can be supported by currently available data. This is desirable since it is sometimes easier to generate engineering estimates of data for the components of an element rather than the entire element. In addition, the existence of advanced models helps justify collection of detailed data. However, with respect to the near-term, it may be more worthwhile to promote data collection than to increase the modeling precision of current reliability techniques.

REFERENCES

(1) Meyer, John F., "Models and Techniques for Evaluating the Effectiveness of Aircraft Computing Systems", NASA Grant NSG 1306, Status Report No. 2, July 1977, NASA CR-145270.

(2) Meyer, John F., "Models and Techniques for Evaluating the Effectiveness of Aircraft Computing Systems", NASA Grant NSG 1306, Status Report No. 3, January 1978, NASA CR-158992.

(3) Meyer, John F., "Models and Techniques for Evaluating the Effectiveness of Aircraft Computing Systems", NASA Grant NSG 1306, Status Report No. 4, July 1978, NASA CR-158993.

(4) Furchtgott, D. G., "METAPHOR (Version 1) Programmer's Guide", NASA Grant NSG 1306, January 1979.

(5) Ballance, R. A., and Meyer, J. F., "Functional Dependence and Its Application to System Evaluation", Proceedings of the 1978 Johns Hopkins Conference on Information Sciences and Systems, Baltimore, MD, March 1978, pp 280-285.

(6) Meyer, J. F., Furchtgott, D. G., and Wu, L. T., "Performability Evaluation of the SIFT Computer", Proc. 1979 Int'l Symposium on Fault-Tolerant Computing, Madison, WI, pp 43-50, June 1979.

(7) Meyer, J. F., "On Evaluating the Performability of Degradable Computing Systems", Proceedings 1978 International Symposium on Fault-Tolerant Computing, Toulouse, France, June 1978, pp 44-49.

(8) Furchtgott, D. G., and Meyer, J. F., "Performability Evaluation of Fault-Tolerant Multiprocessors", GOMAC 1978 Digest of Papers, pp 362-365.

(9) Wu, L. T., and Meyer, J. F., "Phase Models for Evaluating the Performability of Computing Systems", Proceedings 1979 Johns Hopkins Conference on Information Sciences and Systems, Baltimore, MD, March 1979, pp 426-431.

(10) Hagen, E. W., "International Conference on Nuclear Systems Reliability Engineering and Risk Assessment", Nuclear Safety, Vol. 19, No. 1, January-February 1978, pp 38-42.

(11) NRC Risk Assessment Review Group, "Report of the NRC Risk Assessment Review Group on the Reactor Safety Study", Nuclear Safety, Vol. 20, No. 1, January-February 1979, pp 24-26.

(12) Fussell, J. B., Powers, G. J., and Bennetts, R. G., "Fault Trees - A State of the Art Discussion", IEEE Transactions on Reliability, Vol. R-23, April 1974, pp 51-55.

(13) Chamow, Martin F., "Directed Graph Techniques for Fault Tree Analysis", IEEE Transactions on Reliability, Vol. R-27- No. 1, April 1978, pp 7-15.

(14) Pelto, P. J., and Purcell, W. L., "MFAULT: A Computer Program for Analyzing Fault Trees", Battelle Pacific Northwest Laboratories, BNWL-2145, November 1977.

(15) Hopkins, A. L., et al, "FTMP - A Highly Reliable Fault-Tolerant Multiprocessor for Aircraft", Proceedings of the IEEE, Vol. 66, No. 10, October 1978, pp 1221-1239.

APPENDIX A

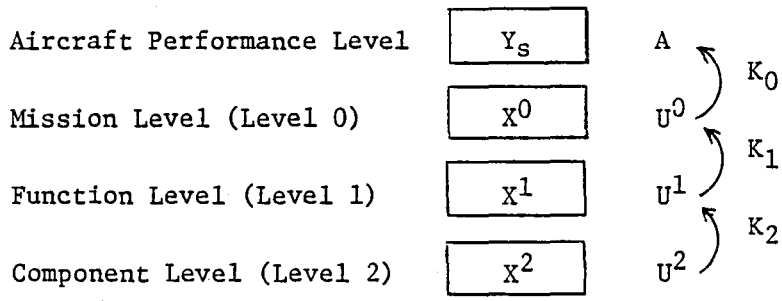
DETAILS OF APPLICATION OF PERFORMABILITY ANALYSIS OF THE DUAL-DUAL PROBLEM

APPENDIX A

DETAILS OF APPLICATION OF PERFORMABILITY ANALYSIS OF THE DUAL-DUAL PROBLEM

This appendix provides details of the application of performability analysis to the dual-dual problem. The problem is described in an earlier section of the report. In addition, the performability analysis solution is summarized in the section entitled "Analysis Results".

Model Hierarchy



Notation

The characteristic function is:

$\gamma = K_0K_1K_2 : U^2 \longrightarrow A$

For $u \in U^2$,

$\gamma(u) = (K_0K_1K_2)(u)$

$= K_0(K_1(K_2(u)))$

The level 0 characteristic function is: $\gamma_0 = K_0$.

The level 1 characteristic function is: $\gamma_1 = K_0K_1$.

The level 2 characteristic function is: $\gamma_2 = \gamma = K_0K_1K_2$.

The probability the mission results in accomplishment level a_n is:

$$P(a_n) = P_r(\gamma^{-1}(a_n)).$$

Accomplishment Set

The set of mission outcomes in the Aircraft Performance Level is the accomplishment set A:

$$A = \{a_0, a_1, a_2\}$$

where

a_0 = safe flight and successful CAT II conditions landing at the primary destination

a_1 = safe flight and landing at alternate destination

a_2 = unsafe flight.

Accomplishment level a_0 requires that the conditions for "safe flight" and "no diversion" in the problem statement are satisfied. Failure to satisfy

the "no diversion" conditions will result in a_1 as long as the "safe flight" conditions are met. If the "safe flight" conditions are not met, then a_2 results whether or not the "no diversion" conditions are satisfied.

Mission Level (Level 0) Model

Let

$$h_1 = \begin{cases} 1 \text{ if no diversion occurs} \\ 0 \text{ otherwise} \end{cases}$$

and

$$h_2 = \begin{cases} 1 \text{ if the flight is safe} \\ 0 \text{ otherwise.} \end{cases}$$

The Level 0 trajectory space is:

$$U^0 = \left\{ \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \;\middle|\; h_i \in \{0,1\} \right\}$$

$$U^0 \xrightarrow{K_0} A$$

The Level 0 inverses are:

$$\gamma_0^{-1}(a_0) = K_0^{-1}(a_0) = \left\{ \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}$$

$$\gamma_0^{-1}(a_1) = K_0^{-1}(a_1) = \left\{ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$$

$$\gamma_0^{-1}(a_2) = K_0^{-1}(a_2) = \left\{ \begin{bmatrix} * \\ 0 \end{bmatrix} \right\}$$

where * indicates "any possible value" (in this case, 0 or 1).

Function Level (Level 1) Model

Let function i (i = 1,2,3,4) be the set of jobs performed by the components in set S_i where:

S_1 = {Radar altimeter, VOR, DME}
S_2 = {DAD, AHRS, INS}
S_3 = {Sensor RT, FCMS, Aft RT}
S_4 = {FCC-1, BIU-1, FCC-2, BIU-2}.

Define

$$f_i = \begin{cases} 2 \text{ if function i meets the "no diversion" requirements} \\ 1 \text{ if funciton i meets the "safe flight" requirements} \\ \quad \text{but not the "no diversion" requirements} \\ 0 \text{ otherwise.} \end{cases}$$

The Level 1 trajectory space is

$$U^1 = \left\{ \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \;\middle|\; f_i \in \{0,1,2\} \right\}$$

$$U^1 \xrightarrow{K_1} U^0 \xrightarrow{K_0} A$$

We need to determine $\gamma_1^{-1} = (K_0K_1)^{-1}$ for all $a_n \in A$. This will be accomplished by characterizing K_1^{-1} for all $u \in U^0$ and then combining K_1^{-1} with $\gamma_0^{-1} = K_0^{-1}$ (from the level 0 model).

Let C_i (i = 1,2) be the mapping defined on U^0 as the projection onto the i^{th} entry:

$$C_i(u) = C_i\left(\begin{bmatrix} h_1 \\ h_2 \end{bmatrix}\right) = h_i.$$

Pictorially we have:

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \in U^1 \xrightarrow{K_1} \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \in U^0 \xrightarrow{K_0} a_n \in A$$

$$\begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \xrightarrow{C_i} h_i$$

The composite map C_iK_1: $U^1 \longrightarrow \{h_i\}$ relates each $v \in U^1$ to a single h_i value. The inverse is defined as follows:

$$(C_iK_1)^{-1}(h_i) = \{v \in U^1 \mid C_i(K_1(v)) = h_i\}.$$

For $u = \begin{bmatrix} h_1 \\ h_2 \end{bmatrix} \in U^0$, the inverse image $K_1^{-1}(u) \in U^1$ can be written as:

$$K_1^{-1}(u) = (C_1K_1)^{-1}(h_1) \cap (C_2K_1)^{-1}(h_2).$$

We will specify $(C_iK_1)^{-1}(h_i)$ for all h_i (i=1,2) and then form $K_1^{-1}(u)$ for each $u \in \gamma_0^{-1}(a_n)$. Finally, since $\gamma_0^{-1}(a_n) \subseteq U^0$, we will form the inverse:

$$\gamma_1^{-1}(a_n) = (K_0K_1)^{-1}(a_n) = \bigcup_{u\ \gamma_0^{-1}(a_n)} K_1^{-1}(u).$$

The inverses for the h_i are as follows:

For $h_1 = 1$ (no diversion):

$$(C_1K_1)^{-1}(1) = \left\{\begin{bmatrix}2\\2\\2\\2\end{bmatrix}\right\}$$

For $h_1 = 0$ (diversion):

$$(C_1K_1)^{-1}(0) = \left\{\begin{bmatrix}0 \text{ or } 1\\*\\*\\*\end{bmatrix}, \begin{bmatrix}2\\0 \text{ or } 1\\*\\*\end{bmatrix}, \begin{bmatrix}2\\2\\0 \text{ or } 1\\*\end{bmatrix}, \begin{bmatrix}2\\2\\2\\0 \text{ or } 1\end{bmatrix}\right\}$$

where * represents "any possible value" (in this case, 0, 1, or 2).

For $h_2 = 1$ (safe flight):

$$(C_2K_1)^{-1}(1) = \left\{\begin{bmatrix}1 \text{ or } 2\\1 \text{ or } 2\\1 \text{ or } 2\\1 \text{ or } 2\end{bmatrix}\right\}$$

For $h_2 = 0$ (unsafe flight):

$$(C_2K_1)^{-1}(0) = \left\{\begin{bmatrix}0\\*\\*\\*\end{bmatrix}, \begin{bmatrix}1 \text{ or } 2\\0\\*\\*\end{bmatrix}, \begin{bmatrix}1 \text{ or } 2\\1 \text{ or } 2\\0\\*\end{bmatrix}, \begin{bmatrix}1 \text{ or } 2\\1 \text{ or } 2\\1 \text{ or } 2\\0\end{bmatrix}\right\}$$

The inverses of the a_n are formed as follows:

Recall $\gamma_0^{-1}(a_o) = \left\{\begin{bmatrix}1\\1\end{bmatrix}\right\}$. Let $u_1 = \left\{\begin{bmatrix}1\\1\end{bmatrix}\right\}$.

$$K_1^{-1}(u_1) = (C_1K_1)^{-1}(1) \cap (C_2K_1)^{-1}(1)$$

$$K_1^{-1}(u_1) = \left\{ \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} \right\}$$

and $\gamma_1^{-1}(a_0) = K_1^{-1}(u_1)$.

Recall $\gamma_0^{-1}(a_1) = \left\{ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$ Let $u_2 = \left\{ \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$

$$K_1^{-1}(u_2) = (C_1K_1)^{-1}(0) \cap (C_2K_1)^{-1}(1)$$

$$K_1^{-1}(u_2) = \left\{ \begin{bmatrix} 1 \\ 1 \text{ or } 2 \\ 1 \text{ or } 2 \\ 1 \text{ or } 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ 1 \text{ or } 2 \\ 1 \text{ or } 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \\ 1 \\ 1 \text{ or } 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \\ 2 \\ 1 \end{bmatrix} \right\}$$

and $\gamma_1^{-1}(a_1) = K_1^{-1}(u_2)$.

Recall $\gamma_0^{-1}(a_2) = \left\{ \begin{bmatrix} * \\ 0 \end{bmatrix} \right\}$. Let $u_3 = \left\{ \begin{bmatrix} * \\ 0 \end{bmatrix} \right\}$.

$$K_1^{-1}(u_3) = (C_1K_1)^{-1}(*) \cap (C_2K_1)^{-1}(0) = (C_2K_1)^{-1}(0)$$

and $\gamma_1^{-1}(a_2) = (C_2K_1)^{-1}(0)$.

Summarizing the Level 1 inverses:

$$\gamma_1^{-1}(a_0) = \left\{ \begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \end{bmatrix} \right\}$$

$$\gamma_1^{-1}(a_1) = \left\{ \begin{bmatrix} 1 \\ 1 \text{ or } 2 \\ 1 \text{ or } 2 \\ 1 \text{ or } 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 1 \\ 1 \text{ or } 2 \\ 1 \text{ or } 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \\ 1 \\ 1 \text{ or } 2 \end{bmatrix}, \begin{bmatrix} 2 \\ 2 \\ 2 \\ 1 \end{bmatrix} \right\}$$

$$\gamma_1^{1}(a_2) = \left\{ \begin{bmatrix} 0 \\ * \\ * \\ * \end{bmatrix}, \begin{bmatrix} 1 \text{ or } 2 \\ 0 \\ * \\ * \end{bmatrix}, \begin{bmatrix} 1 \text{ or } 2 \\ 1 \text{ or } 2 \\ 0 \\ * \end{bmatrix}, \begin{bmatrix} 1 \text{ or } 2 \\ 1 \text{ or } 2 \\ 1 \text{ or } 2 \\ 0 \end{bmatrix} \right\}$$

Component Level (Level 2) Model

Let

x_{ij} = number of units of component i which are fault-free in Phase j (j=1,2).

The component subscripts i (i=1,2,...,13) and the domains of the x_{ij} are defined in Table 1. The Level 2 trajectory space is the set of 13 by 2 matrices.

$$U^2 = \{[x_{ij}] \mid i = 1, 2, \ldots, 13 \text{ and } j = 1,2\}.$$

Rows of $[x_{ij}]$ correspond to components and columns correspond to phases. U^2 can be functionally related to the accomplishment set A as follows:

$$U^2 \xrightarrow{K_2} U^1 \xrightarrow{\gamma_1 = K_0 K_1} A$$

The inverses $\gamma_1^{-1}(a_n)$ were specified in the preceding section. We now need to specify $\gamma_2^{-1} = (\gamma_1 K_2)^{-1}$ for all $a_n \in A$. This will be accomplished by characterizing K_2^{-1} for all $v \in U^1$ and then combining K_2^{-1} with γ_1^{-1}.

Let α_i (i = 1,2,3,4) be the mapping defined on U^1 as the projection onto the i^{th} entry:

$$\alpha_i(v) = \alpha_i \left(\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \right) = f_i.$$

Pictorially we have:

$$\begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ x_{13,1} & x_{13,2} \end{bmatrix} \in U^2 \xrightarrow{K_2} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \in U^1 \xrightarrow{\gamma_1} A$$

$$\downarrow \alpha_i$$

$$f_i$$

Then

$$(\alpha_i K_2)^{-1}(f_i) = \{w \in U^2 \mid \alpha_i\left(K_2(w)\right) = f_i\}$$

and for $v = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \in U^1$,

$$K_2^{-1}(v) = \bigcap_{i=1}^{4} (\alpha_i K_2)^{-1}(f_i).$$

According to the descriptions of performability analysis, the next steps are:

- Specify the sets $(\alpha_i K_2)^{-1}(f_i)$ for all f_i (i=1,2,3,4)
- For each $v \in \gamma_1^{-1}(a_n)$, determine $K_2^{-1}(v)$ using the intersections of the $(\alpha_i K_2)^{-1}(f_i)$
- Compute $\gamma^{-1}(a_n) = \bigcup_{v \in \gamma_1^{-1}(a_n)} K_2^{-1}(v)$
- Compute $P(a_n) = \Pr\left(\gamma^{-1}(a_n)\right)$

Each $\gamma^{-1}(a_n)$ is a set of 13 by 2 $[x_{ij}]$ matrices in U^2. The trajectory space U^2 has the structure Q x Q where Q is the state space of 13-dimensional vectors:

$$Q = \left\{ \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_{13} \end{bmatrix} \right\}$$

where each x_i corresponds to x_{ij} in Table 1 (i.e., the phase subscript j is omitted).

Q is a space of dimension 13. While it is conceptually possible to find inverses of elements of U^1 in U^2 = Q x Q and to develop the probability transition matrices, the practical aspects of such an undertaking are prohibitive.

We can proceed by decomposing Q into mutually independent subspaces. Two subspaces are independent if, for all q in the space, the values of the components of q in each subspace are not impacted by the values of the com-

TABLE 1. COMPONENT VARIABLES x_{ij} FOR THE LEVEL 2 MODEL

Component	i	Domain of x_{ij}
Radar Altimeter	1	0,1
VOR	2	0,1,2
DME	3	0,1,2
DAD	4	0,1,2
AHRS	5	0,1
INS	6	0,1
Sensor RT	7	0,1,2
FCMS	8	0,1,2
Aft RT	9	0,1,2
FCC-1	10	0,1
BIU-1	11	0,1,2
FCC-2	12	0,1
BIU-2	13	0,1,2

ponents of q in the other subspace. By virtue of their independence, we can find inverses and compute probabilities in each subspace. From a practical point of view, the dimension of each subspace will be manageable.

Decompose Q into four subspaces, denoted by Q_i, and defined as follows:

$$Q_1 = \left\{ \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \right\}$$

$$Q_2 = \left\{ \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} \right\}$$

$$Q_3 = \left\{ \begin{bmatrix} x_7 \\ x_8 \\ x_9 \end{bmatrix} \right\}$$

$$Q_4 = \left\{ \begin{bmatrix} x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \end{bmatrix} \right\} .$$

The components for each subspace were chosen to assure mutually independent subspaces. Table 2, (which is based on Table 2 of the problem statement), formed the basis of selection. The DAD, AHRS, and INS interact and are grouped to form Q_2. The FCCs and BIUs interact and are grouped to form Q_4. The remaining components are all independent with respect to Q_2, Q_4, and each other. For convenience, they are grouped into two subspaces, each of dimension three. Note that the components in subspace i correspond to the components comprising function i.

Since the f_i are independent in terms of their contributions to the mission outcome, the performability analysis can be completed according to the following steps:

- Specify $(\alpha_i K_2)^{-1}(f_i)$ in the subspace Q_i
- Compute $Pr(F_i=f_i) = Pr\ [(\alpha_i K_2)^{-1}(f_i)]$ using equation 5 in Reference 1 (where F_i represents function i)
- For each $v = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \epsilon \gamma_1^{-1}(a_n)$, compute $\prod_{i=1}^{4} Pr(F_i=f_i)$

TABLE 2. COMPONENT REQUIREMENTS FOR MISSION PERFORMANCE LEVELS

Component	MINIMUM COMPONENT REQUIREMENTS: Safe Flight (both phases)	Initiate CAT II Landing (T=73 min)	Complete CAT II Landing (T=75 min)
Radar Alt.		1	1
Digital Air Data	1	2	(1 / 1) or 1
AHRS	1 or 1	1 or 1	
INS			
VOR		2	1
DME		1	
Sensor RT	1	2	1
PU-I	1		1
PU-II		2	
FCMS	1	2	1
Aft RT	1	2	1

where

PU = processing unit
PU-I: one FCC with one associated BIU
PU-II: one FCC with both associated BIUs

- Sum the products $\prod_{i=1}^{4} \Pr(F_i=f_i)$ for all $v \in \gamma_1^{-1}(a_n)$ to obtain $P(a_n)$.

The procedure and computations for subspace Q_1 are explained in some detail. Since subspaces Q_2, Q_3, and Q_4 are treated analogously, explanations are omitted from those computations.

Subspace Q_1. The inverse images $(\alpha_1 K_2)^{-1}(f_1)$ can be completely specified in terms of the trajectory subspace $U_1{}^2 = Q_1 \times Q_1$ where $Q_1 = \left\{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right\}$.

Using Table 1 we can form the following table:

	No Diversion		Safe Flight
	Phase 1	Phase 2	Phases 1 and 2
x_1	1	1	*
x_2	2	1 or 2	*
x_3	1 or 2	*	*

where * represents "any possible value". From this table we can specify the $(\alpha_1 K_2)^{-1}(f_1)$:

$$(\alpha_1 K_2)^{-1}(2) = \left\{\begin{bmatrix} 1 & 1 \\ 2 & 1 \text{ or } 2 \\ 1 \text{ or } 2 & * \end{bmatrix}\right\}$$

$$(\alpha_1 K_2)^{-1}(1) = \left\{\begin{bmatrix} 1 & 0 \\ 2 & * \\ 1 \text{ or } 2 & * \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 2 & 0 \\ 1 \text{ or } 2 & * \end{bmatrix}, \begin{bmatrix} 0 & * \\ * & * \\ * & * \end{bmatrix}, \begin{bmatrix} 1 & * \\ 0 \text{ or } 1 & * \\ * & * \end{bmatrix}, \begin{bmatrix} 1 & * \\ 2 & * \\ 0 & * \end{bmatrix}\right\}$$

$$(\alpha_1 K_2)^{-1}(0) = \emptyset.$$

The states in the subspace Q_1 are diagrammed in Figure 1. We can write the elements of the above sets as Cartesian sets in terms of the state numbers:

$$\left\{\begin{bmatrix} 1 & 1 \\ 2 & 1 \text{ or } 2 \\ 1 \text{ or } 2 & * \end{bmatrix}\right\} = \{1, 2\} \times \{1, 2, 3, 4, 5, 6\} = V_1$$

$$\left\{\begin{bmatrix} 1 & 0 \\ 2 & * \\ 1 \text{ or } 2 & * \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 2 & 0 \\ 1 \text{ or } 2 & * \end{bmatrix}\right\} = \{1, 2\} \times \{7\} = V_2$$

$$\left\{\begin{bmatrix} 0 & * \\ * & * \\ * & * \end{bmatrix}, \begin{bmatrix} 1 & * \\ 0 \text{ or } 1 & * \\ * & * \end{bmatrix}, \begin{bmatrix} 1 & * \\ 2 & * \\ 0 & * \end{bmatrix}\right\} = \{3, 4, 5, 6, 7\} \times Q = V_3$$

Hence,

$$(\alpha_1 K_2)^{-1}(2) = V_1$$
$$(\alpha_1 K_2)^{-1}(1) = V_2 \cup V_3$$
$$(\alpha_1 K_2)^{-1}(0) = \emptyset.$$

From Reference 1 we have, for each V_i,

$$Pr(V_i) = I(0) \cdot P_1 \cdot G_{V_i,1} \cdot P_2 \cdot G_{V_i,2} \cdot F$$

where

$I(0) = [1\ 0\ 0\ 0\ 0\ 0\ 0]$

$F = [\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^t$ (t indicates "transpose")

P_k = intraphase transition matrix for phase k

$G_{V_n,k}$ = characteristic matrix for V_n and phase k.

The intraphase transition matrix is $P_k = [P_k(i,j)]$

where

$P_k(i,j)$ = Pr (system ends phase k in state j | system begins phase k in state i).

Each $P_k(i,j)$ is expressed in terms of

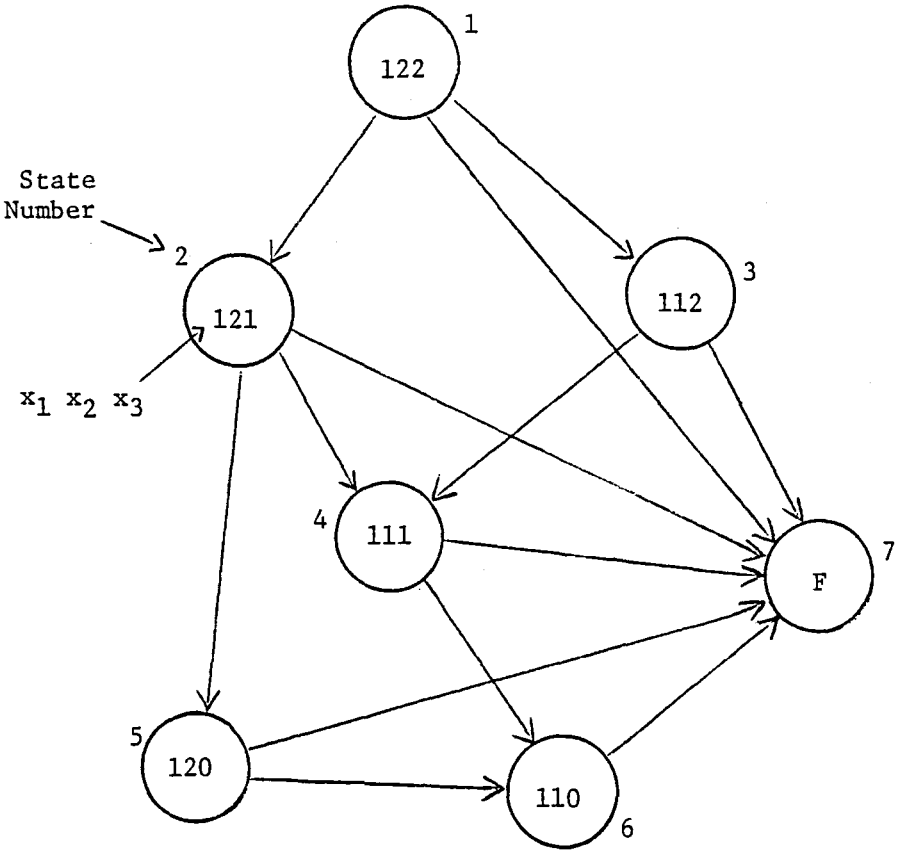
$P_n = \exp(-\lambda_n t_k) = P_r$ (component of type n does not fail in time t_k)

$q_n = 1 - P_n$

where

λ_n = failure rate of a type n component

t_k = duration of phase k.



$Q = \{1,2,3,4,5,6,7\}$

$U = Q_1 \times Q_1$

FIGURE 1. STATE DIAGRAM FOR THE SUBSPACE Q_1

Figure 2 presents P_k for subspace Q_1. Since the same model (i.e., subspace) is used in each phase, the only difference between P_1 and P_2 is in their durations.

The characteristic matrix for phase k of the Cartesian set V_n is $G_{V_n,k} = [G_{V_n,k}(i,j)]$ where

$$G_{V_n,k}(i,j) = \begin{cases} 1 \text{ if } i=j \text{ and } i \in \text{kth state set of } V_n \\ 0 \text{ otherwise} \end{cases}$$

The role of $G_{V_n,k}$ is to select the output states of the intraphase transition matrix P_k which correspond to the Caresian set V_n. Multiplying P_k by $G_{V_n,k}$ puts zeros in all columns of P_k except those corresponding to the phase k end states of V_n.

The symbolic computations to derive $Pr(V_n)$ in terms of the $P_k(i,j)$ are as follows:

$$V_1 = \{1,2\} \times \{1,2,3,4,5,6\}$$

$$G_{V_1,1} = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 0 & & & & \\ & & & 0 & & & \\ & & & & 0 & & \\ & & & & & 0 & \\ & & & & & & 0 \end{bmatrix} \qquad G_{V_1,2} = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 1 & & & & \\ & & & 1 & & & \\ & & & & 1 & & \\ & & & & & 1 & \\ & & & & & & 0 \end{bmatrix}$$

$$I(0) \cdot P_{V,1} = [P_1(1,1) \;\; P_1(1,2) \;\; P_1(1,3) \;\; P_1(1,4) \;\; P_1(1,5) \;\; P_1(1,6) \;\; P_1(1,7)]$$

$$I(0) \cdot P_{V,1} \cdot G_{V_1,1} = [P_1(1,1) \;\; P_1(1,2) \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]$$

$$P_{V,2} \cdot G_{V_1,2} \cdot F =$$

$$\begin{bmatrix} P_2(1,1) + P_2(1,2) + P_2(1,3) + P_2(1,4) + P_2(1,5) + P_2(1,6) \\ P_2(2,2) + P_2(2,4) + P_2(2,5) + P_2(2,6) \\ P_2(3,3) + P_2(3,4) + P_2(3,6) \\ P_2(4,4) + P_2(4,6) \\ P_2(5,5) + P_2(5,6) \\ P_2(6,6) \\ 0 \end{bmatrix}$$

$$P_k = \begin{bmatrix} P_1P_2^2P_3^2 & 2P_1P_2^2P_3q_3 & 2P_1P_2q_2P_3^2 & 4P_1P_2q_2P_3q_3 & P_1P_2^2q_3^2 & 2P_1P_2q_2q_3^2 & q_1 + P_1q_2^2 \\ 0 & P_1P_2^2P_3 & 0 & 2P_1P_2q_2P_3 & P_1P_2^2q_3 & 2P_1P_2q_2q_3 & q_1 + P_1q_2^2 \\ 0 & 0 & P_1P_2P_3^2 & 2P_1P_2P_3q_3 & 0 & P_1P_2q_3^2 & q_1 + P_1q_2 \\ 0 & 0 & 0 & P_1P_2P_3 & 0 & P_1P_2q_3 & q_1 + P_1q_2 \\ 0 & 0 & 0 & 0 & P_1P_2^2 & 2P_1P_2q_2 & q_1 + P_1q_2^2 \\ 0 & 0 & 0 & 0 & 0 & P_1P_2 & q_1 + P_1q_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

FIGURE 2: INTRAPHASE TRANSITION MATRIX FOR SUBSPACE Q_1. SUBSCRIPTS ON THE P_i TO INDICATE PHASE k ARE OMITTED. $P_i = \exp[-\lambda_i t_k]$ $q_i = 1 - P_i$

$$P_r(V_1) = I(0) \cdot P_{v,1} \cdot G_{V_1,1} \cdot P_{V,2} \cdot G_{V_1,2} \cdot F$$

$$= P_1(1,1) \cdot \sum_{j=1}^{6} P_2(1,j)$$

$$+ P_1(1,2)[P_2(2,2) + P_2(2,4) + P_2(2,5) + P_2(2,6)]$$

Next, $V_2 = \{1,2\} \times \{7\}$

$$G_{V_2,1} = \begin{bmatrix} 1 & & & & & & \\ & 1 & & & & & \\ & & 0 & & & & \\ & & & 0 & & & \\ & & & & 0 & & \\ & & & & & 0 & \\ & & & & & & 0 \end{bmatrix} \qquad G_{V_2,2} = \begin{bmatrix} 0 & & & & & & \\ & 0 & & & & & \\ & & 0 & & & & \\ & & & 0 & & & \\ & & & & 0 & & \\ & & & & & 0 & \\ & & & & & & 1 \end{bmatrix}$$

$$I(0) \cdot P_{V,1} \cdot G_{V_2,1} = [P_1(1,1) \quad P_1(1,2) \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$P_{V,2} \cdot G_{V_2,2} \cdot F = \begin{bmatrix} P_2(1,7) \\ P_2(2,7) \\ P_2(3,7) \\ P_2(4,7) \\ P_2(5,7) \\ P_2(6,7) \\ P_2(7,7) \end{bmatrix}$$

$$Pr(V_2) = P_1(1,1)P_2(1,7) + P_1(1,2)P_2(2,7)$$

Next, $V_3 = \{3,4,5,6,7\} \times \{1,2,3,4,5,6,7\}$

$$G_{V_3,1} = \begin{bmatrix} 0 & & & & & \\ & 0 & & & & \\ & & 1 & & & \\ & & & 1 & & \\ & & & & 1 & \\ & & & & & 1 \end{bmatrix} \qquad G_{V_3,2} = I_7$$

$$I(0) \cdot P_{V,1} \cdot G_{V_3,1} = [0 \quad 0 \quad P_1(1,3) \quad P_1(1,4) \quad P_1(1,5) \quad P_1(1,6) \quad P_1(1,7)]$$

$$P_{V,2} \cdot G_{V_3,2} \cdot F = P_{V,2} F = [1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1]^t$$

$$P_r(V_3) = P_1(1,3) + P_1(1,4) + P_1(1,5) + P_1(1,6) + P_1(1,7)$$

From Table 1 of the problem statement:

	Component	MTBF(hrs)	λ_n
1	Radar Alt.	700	.001429
2	VOR	1000	.0010
3	DME	1000	.0010

The numerical inputs for the P_n, q_n are as follows:

Component	Phase 1		Phase 2	
	P_n	q_n	P_r	q_r
1	.99826341	1.73659 E-3	.99995238	4.76180 E-5
2	.99878407	1.21593 E-3	.99996667	3.33328 E-5
3	.99878407	1.21593 E-3	.99996667	3.33328 E-5

where "E-a" represents 10^{-a}.

Using the above values and Figure 2, the following values are computed:

$P_1(1,1) = .99341698$
$P_1(1,2) = 2.41879\ E-3$
$P_1(1,3) = 2.41879\ E-3$
$P_1(1,4) = 5.88929\ E-6$
$P_1(1,5) = 1.47232\ E-6$
$P_1(1,6) = 3.58483\ E-9$
$P_1(1,7) = 1.73806\ E-3$

$P_2(1,1) = .99981907$
$P_2(1,2) = 6.66558\ E-5$
$P_2(1,3) = 6.66558\ E-5$
$P_2(1,4) = 4.44379\ E-9$
$P_2(1,5) = 1.11095\ E-9$
$P_2(1,6) = 7.4\ \ E-14$
$P_2(1,7) = 4.76191\ E-5$

$P_2(2,2) = .99985240$

$P_2(2,4) = 6.66580\ E-5$

$P_2(2,5) = 3.33290\ E-5$

$P_2(2,6) = 2.22197\ E-9$

$P_2(2,7) = 4.76191\ E-5$

$P_2(3,3) = .99985239$

$P_2(3,4) = 6.6657981\ E-5$

$P_2(3,6) = 1.11098\ E-9$

$P_2(3,7) = 8.0949212\ E-5$

$P_2(4,4) = .99988572$

$P_2(4,6) = 3.333010\ E-5$

$P_2(4,7) = 8.094921\ E-5$

$P_2(5,5) = .99988572$

$P_2(5,6) = 6.6660203\ E-5$

$P_2(5,7) = 4.7619111\ E-5$

$P_2(6,6) = .99991905$

$P_2(6,7) = 8.0949212\ E-5$

Substituting the $P_k(i,j)$ values into the $P_r(V_n)$ expressions yields:

$P_r(V_1) = .995788$

$P_r(V_2) = 4.74208\ E-5$

$P_r(V_3) = 4.16422\ E-3.$

Finally, we have:

$P_r(F_1=2) = P_r(V_1) = .995788$

$P_r(F_1=1) = P_r(V_2) + P_r(V_3) = 4.21164\ E-3$

$P_r(F_1=0) = P_r(\emptyset) = 0.$

Subspace Q_2. Each $(\alpha_2 k_2)^{-1}(f_2)$, $f_2 \in \{0,1,2\}$, is a union of Cartesian sets in $U_2{}^2 = Q_2 \times Q_2$ where

$$Q_2 = \left\{ \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix} \right\}$$

From Table 1 we have:

	No Diversion		Safe Flight
	Phase 1	Phase 2	Phases 1,2
x_4	$\begin{pmatrix} 1 \\ * \end{pmatrix}$ or $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1,2 \\ 1 \\ * \end{pmatrix} \begin{pmatrix} 1,2 \\ 0 \\ 1 \end{pmatrix} \begin{pmatrix} 0 \\ * \\ 1 \end{pmatrix}$	$\begin{pmatrix} 1,2 \\ 1 \\ * \end{pmatrix} \begin{pmatrix} 1,2 \\ 0 \\ 1 \end{pmatrix}$
x_5			
x_6			

Each $q = \begin{bmatrix} x_4 \\ x_5 \\ x_6 \end{bmatrix}$ is a state in Q_2. Figure 3 displays the state diagram for Q_2.

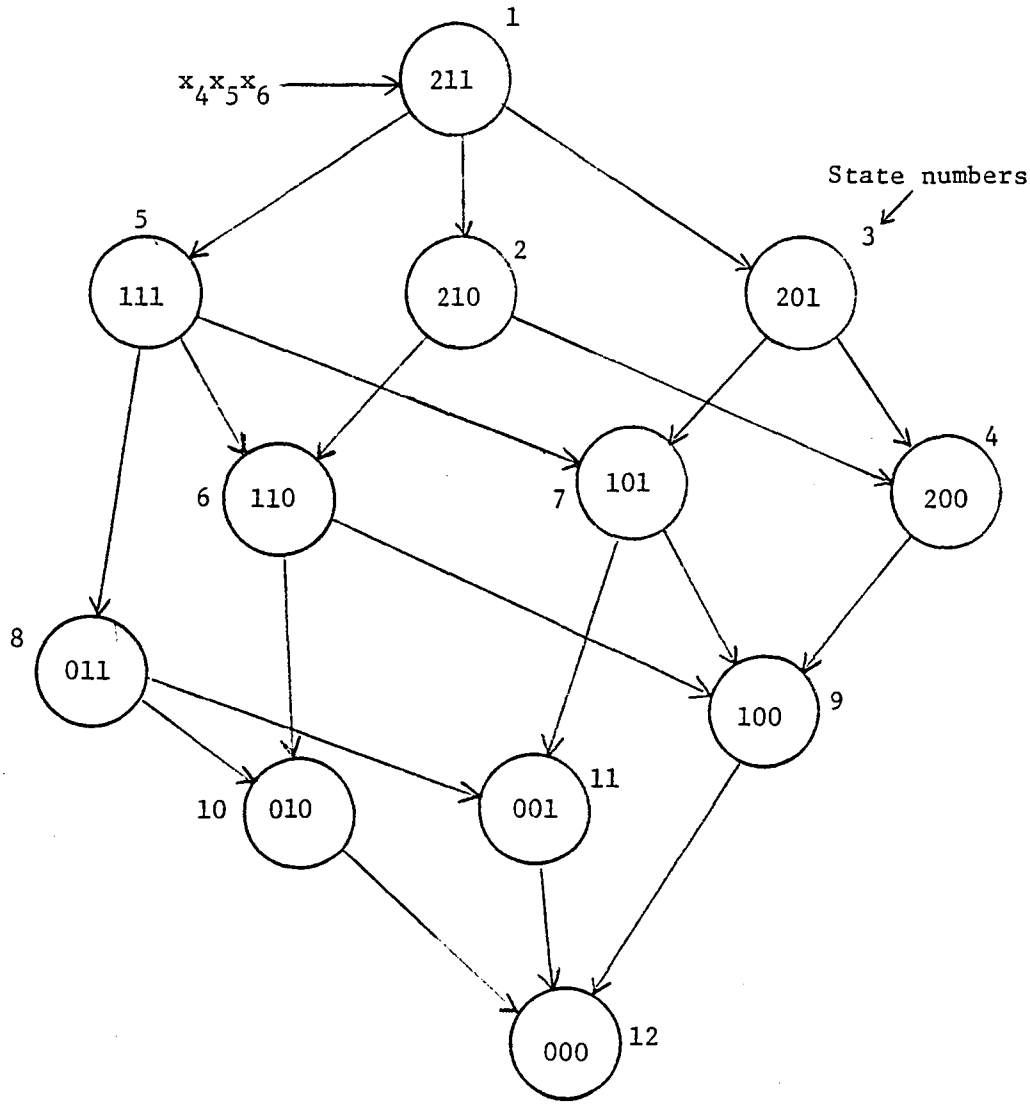


FIGURE 3. STATE TRANSITION DIAGRAM FOR Q_2.

$Q_2 = \{1,2,3,4,5,6,7,8,9,10,11,12\}$

$U_2^2 = Q_2 \times Q_2$

A state number is associated with each $q \in Q_2$. Using the state numbers we can rewrite the above table as:

Condition	No Diversion		Safe Flight
	Phase 1	Phase 2	Phases 1,2
States	1,2,3	1,2,3,5, 6,7,8,11	1,2,3, 5,6,7

The $(\alpha_2 K_2)^{-1}(f_2)$ sets can now be written in terms of Cartesian sets of states in Q_2:

$(\alpha_2 K_2)^{-1}(2) = V_1$ where

V_1 = {no diversion, phase 1} x {no diversion, phase 2}

V_1 = {1,2,3} x {1,2,3,5,6,7,8,11}

$(\alpha_2 K_2)^{-1}(1) = V_2 \cup V_3$ where

V_2 = {no diversion, phase 1} x {diversion and safe flight, phase 2}

V_2 = {1,2,3} x $\emptyset$

and

V_3 = {diversion and safe flight, phase 1} x {safe flight, phase 2}

V_3 = {5,6,7} x {1,2,3,5,6,7}

$(\alpha_2 K_2)^{-1}(0) = V_4 \cup V_5 \cup V_6$ where

V_4 = {unsafe, phase 1} x {all states, phase 2}

V_4 = {4,8,9,10,11,12} x Q_2

and

V_5 = {no diversion and safe, phase 1} x {unsafe, phase 2}

V_5 = {1,2,3} x {4,9,10,12}

and

V_6 = {diversion and safe, phase 1} x {unsafe, phase 2}

V_6 = {5,6,7} x {4,8,9,10,11,12}.

Figure 4 shows the intraphase transition matrix. The symbolic computations for $P_r(V_i)$ are as follows:

$$P_{V,k} = \begin{bmatrix} P(1,1) & P(1,2) & P(1,3) & P(1,4) & P(1,5) & P(1,6) & P(1,7) & P(1,8) & P(1,9) & P(1,10) & P(1,11) & P(1,12) \\ 0 & P(2,2) & 0 & P(2,4) & 0 & P(2,6) & 0 & 0 & P(2,9) & P(2,10) & 0 & P(2,12) \\ 0 & 0 & P(3,3) & P(3,4) & 0 & 0 & P(3,7) & 0 & P(3,9) & 0 & P(3,11) & P(3,12) \\ 0 & 0 & 0 & P(4,4) & 0 & 0 & 0 & 0 & P(4,9) & 0 & P(4,11) & P(4,12) \\ 0 & 0 & 0 & 0 & P(5,5) & P(5,6) & P(5,7) & P(5,8) & P(5,9) & P(5,10) & P(5,11) & P(5,12) \\ 0 & 0 & 0 & 0 & 0 & P(6,6) & 0 & 0 & P(6,9) & P(6,10) & 0 & P(6,12) \\ 0 & 0 & 0 & 0 & 0 & 0 & P(7,7) & 0 & P(7,9) & 0 & P(7,11) & P(7,12) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & P(8,8) & 0 & P(8,10) & P(8,11) & P(8,12) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P(9,9) & 0 & 0 & P(9,12) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P(10,10) & 0 & P(10,12) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & P(11,11) & P(11,12) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

FIGURE 4. INTRAPHASE TRANSITION MATRIX FOR $V \subseteq Q_2$ AND PHASE k

$$V_1 = \{1,2,3\} \times \{1,2,3,5,6,7,8,11\}$$

$$I(0) \cdot P_{V,1}\, G_{V_1,1} = [P_1(1,1)\ \ P_1(1,2)\ \ P_1(1,3)\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]$$

$$P_{V,2} \cdot G_{V_1,2}F =$$

$$\begin{bmatrix}
P_2(1,1) + P_2(1,2) + P_2(1,3) + P_2(1,5) + P_2(1,6) + P_2(1,7) + P_2(1,8) + P_2(1,11) \\
P_2(2,2) + P_2(2,6) \\
P_2(3,3) + P_2(3,7) + P_2(3,11) \\
0 \\
P_2(5,5) + P_2(5,6) + P_2(5,7) + P_2(5,8) + P_2(5,11) \\
P_2(6,6) \\
P_2(7,7) + P_2(7,11) \\
P_2(8,8) + P_2(8,11) \\
0 \\
0 \\
P_2(11,11) \\
0
\end{bmatrix}$$

$$\begin{aligned}
Pr(V_1) &= I(0) \cdot P_{V,1} \cdot G_{V_1,1} \cdot P_{V,2} \cdot G_{V_1,2} \cdot F \\
&= P_1(1,1)\,[P_2(1,1) + P_2(1,2) + P_2(1,3) + P_2(1,5) + P_2(1,6) + P_2(1,7) + P_2(1,8) \\
&\quad + P_2(1,11)] \\
&\quad + P_1(1,2)[P_2(2,2) + P_2(2,6)] \\
&\quad + P_1(1,3)[P_2(3,3) + P_2(3,7) + P_2(3,11)]
\end{aligned}$$

$Pr(V_2) = 0$ (since $P_r(\emptyset) = 0$).

Next, $V_3 = \{5,6,7\} \times \{1,2,3,5,6,7\}$

$$I(0) \cdot P_{V,1} \cdot G_{V_3,1} = [\,0\ \ 0\ \ 0\ \ 0\ \ P_1(1,5)\ \ P_1(1,6)\ \ P_1(1,7)\ \ 0\ \ 0\ \ 0\ \ 0\ \ 0]$$

$$P_{V,3} \cdot G_{V_2,3} \cdot F$$

$$= \begin{bmatrix} P_2(1,1) + P_2(1,2) + P_2(1,3) + P_2(1,5) + P_2(1,6) + P_2(1,7) \\ P_2(2,2) + P_2(2,6) \\ P_2(3,3) + P_2(3,7) \\ 0 \\ P_2(5,5) + P_2(5,6) + P_2(5,7) \\ P_2(6,6) \\ P_2(7,7) \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$Pr(V_3) = P_1(1,5)[P_2(5,5) + P_2(5,6) + P_2(5,7)] + P_1(1,6)P_2(6,6) + P_1(1,7)P_2(7,7)$$

Next, $V_4 = \{4,8,9,10,11,12\} \times Q$

$$I(0) \cdot P_{V,1} \cdot G_{V_4,1}$$

$$= [0 \;\; 0 \;\; 0 \;\; P_1(1,4) \;\; 0 \;\; 0 \;\; 0 \;\; P_1(1,8) \;\; P_1(1,9) \;\; P_1(1,10) \;\; P_1(1,11) \;\; P_1(1,12)]$$

$$P_{V,2} \cdot G_{V_4,2} \cdot F = [1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1 \;\; 1]^t$$

$$Pr(V_4) = P_1(1,4) + P_1(1,8) + P_1(1,9) + P_1(1,10) + P_1(1,11) + P_1(1,12)$$

Next, $V_5 = \{1,2,3\} \times \{4,9,10,12\}$

$$I(0) \cdot P_{V,1} \cdot G_{V_5,1} = [P_1(1,1) \;\; P_1(1,2) \;\; P_1(1,3) \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0 \;\; 0]$$

$$P_{V,2} \cdot G_{V_5,2} \cdot F = \begin{bmatrix} P_2(1,4) + P_2(1,9) + P_2(1,10) + P_2(1,12) \\ P_2(2,4) + P_2(2,9) + P_2(2,10) + P_2(2,12) \\ P_2(3,4) + P_2(3,9) + P_2(3,12) \\ P_2(4,4) + P_2(4,9) + P_2(4,12) \\ P_2(5,9) + P_2(5,10) + P_2(5,12) \\ P_2(6,9) + P_2(6,10) + P_2(6,12) \\ P_2(7,9) + P_2(7,12) \\ P_2(8,10) + P_2(8,12) \\ P_2(9,9) + P_2(9,12) \\ P_2(10,10) + P_2(10,12) \\ P_2(11,12) \\ 1 \end{bmatrix}$$

$$\begin{aligned} Pr(V_5) = {} & P_1(1,1)[P_2(1,4) + P_2(1,9) + P_2(1,10) + P_2(1,12)] \\ & + P_1(1,2)[P_2(2,4) + P_2(2,9) + P_2(2,10) + P_2(2,12)] \\ & + P_1(1,3)[P_2(3,4) + P_2(3,9) + P_2(3,11) + P_2(3,12) \end{aligned}$$

Next, $V_6 = \{5,6,7\} \times \{4,8,9,10,11,12\}$

$$I(0) \cdot P_{V,1} \cdot G_{V_6,1} = [0 \quad 0 \quad 0 \quad 0 \quad P_1(1,5) \quad P_1(1,6) \quad P_1(1,7) \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]$$

$$P_{V,2} \cdot G_{V_6,2} \cdot F = \begin{bmatrix} P_2(1,4) + P_2(1,8) + P_2(1,9) + P_2(1,10) + P_2(1,11) + P_2(1,12) \\ P_2(2,4) + P_2(2,9) + P_2(2,10) + P_2(2,12) \\ P_2(3,4) + P_2(3,9) + P_2(3,11) + P_2(3,12) \\ P_2(4,4) + P_2(4,9) + P_2(4,11) + P_2(4,12) \\ P_2(5,8) + P_2(5,9) + P_2(5,10) + P_2(5,11) + P_2(5,12) \\ P_2(6,9) + P_2(6,10) + P_2(6,12) \\ P_2(7,9) + P_2(7,11) + P_2(7,12) \\ P_2(8,8) + P_2(8,10) + P_2(8,11) + P_2(8,12) \\ P_2(9,9) + P_2(9,12) \\ P_2(10,10) + P_2(10,12) \\ P_2(11,11) + P_2(11,12) \\ 1 \end{bmatrix}$$

$$Pr(V_6) = P_1(1,5)[P_2(5,8) + P_2(5,9) + P_2(5,10) + P_2(5,11) + P_2(5,12)]$$
$$+ P_1(1,6)[P_2(6,9) + P_2(6,10) + P_2(6,12)]$$
$$+ P_1(1,7)[P_2(7,9) + P_2(7,11) + P_2(7,12)]$$

Using the MTBF data from the problem statement:

n	Phase 1		Phase 2	
	P_n	q_n	P_n	q_n
4	.99939185	6.08148 E-4	.99998333	1.66666 E-5
5	.99848032	1.51968 E-3	.99995833	4.16659 E-5
6	.99595266	4.04734 E-3	.99988890	1.11105 E-4

The $P_k(i,j)$ computations are:

$P_1(1,1) = P_4^2P_5P_6 = .99322997$

$P_1(1,2) = P_4^2P_5q_6 = 4.036278$ E-3

$P_1(1,3) = P_4^2q_5P_6 = 1.511687$ E-3

$P_1(1,4) = P_4^2q_5q_6 = 6.14318$ E-6

$P_1(1,5) = 2P_4q_4P_5P_6 = 1.208798$ E-3

$P_1(1,6) = 2P_4q_4P_5q_6 = 4.91230$ E-6

$P_1(1,7) = 2P_4q_4q_5P_6 = 1.83978$ E-6

$P_1(1,8) = q_4^2P_5P_6 = 3.67788$ E-7

$P_1(1,9) = 2P_4q_4q_5q_6 = 7.47647$ E-9

$P_1(1,10) = q_4^2P_5q_6 = 1.49461$ E-9

$P_1(1,11) = q_4^2q^5P_6 = 5.5977$ E-10

$P_1(1,12) = q_4^2q_5q_6 = 2.27$ E-12

$P_2(1,1) = .99981390$

$P_2(1,2) = 1.11097$ E-4

$P_2(1,3) = 4.16599$ E-5

$P_2(1,4) = 4.62914$ E-9

$P_2(1,5) = 3.33276$ E-5

$P_2(1,6) = 3.70327$ E-9

$P_2(1,7) = 1.389$ E-9

$P_2(1,8) = 2.78$ E-10

$P_2(1,9) = 1.5$ E-13

$P_2(1,10) = 3.1$ E-14

$P_2(1,11) = 1.2$ E-14

$P_2(1,12) = 1.3$ E-18

$P_2(2,2) = P_4^2 P_5 = .99992499$

$P_2(2,4) = P_4^2 q_5 = 4.16645$ E-5

$P_2(2,6) = 2P_4 q_4 P_5 = 3.33313$ E-5

$P_2(2,9) = 2P_4 q_4 q_5 = 1.38883$ E-9

$P_2(2,10) = q_4^2 P_5 = 2.78$ E-10

$P_2(2,12) = q_4^2 q_5 = 1.16$ E-14

$P_2(3,3) = P_4^2 P_6 = .99985556$

$P_2(3,4) = P_4^2 q_6 = 1.11101$ E-4

$P_2(3,7) = 2P_4 q_4 P_6 = 3.33289$ E-5

$P_2(3,9) = 2P_4 q_4 q_6 = 3.7034$ E-9

$P_2(3,11) = q_4^2 P_6 = 2.778$ E-10

$P_2(3,12) = q_4^2 q_6 = 3.1$ E-14

$P_2(4,4) = P_4^2 = .99996666$

$P_2(4,9) = 2P_4 q_4 = 3.33326$ E-5

$P_2(4,12) = q_4^2 = 2.78$ E-10

$P_2(5,5) = P_4 P_5 P_6 = .99983056$

$P_2(5,6) = P_4 P_5 q_6 = 1.11099$ E-4

$P_2(5,7) = P_4 q_5 P_6 = 4.16606$ E-5

$P_2(5,8) = q_4 P_5 P_6 = 1.66641$ E-5

$P_2(5,9) = P_4 q_5 q_6 = 4.62921\ E\text{-}9$

$P_2(5,10) = q_4 P_5 q_6 = 1.852\ E\text{-}9$

$P_2(5,11) = q_4 q_5 P_6 = 6.94\ E\text{-}10$

$P_2(5,12) = q_4 q_5 q_6 = 7.7\ E\text{-}14$

$P_2(6,6) = P_4 P_5 = .99994166$

$P_2(6,9) = P_4 q_5 = 4.16652\ E\text{-}5$

$P_2(6,10) = q_4 P_5 = 1.66659\ E\text{-}5$

$P_2(6,12) = q_4 q_5 = 6.9442\ E\text{-}10$

$P_2(7,7) = P_4 P_6 = .99987223$

$P_2(7,9) = P_4 q_6 = 1.11103\ E\text{-}4$

$P_2(7,11) = q_4 P_6 = 1.66647\ E\text{-}5$

$P_2(7,12) = q_4 q_6 = 1.852\ E\text{-}9$

$P_2(8,8) = P_5 P_6 = .99984723$

$P_2(8,10) = P_5 q_6 = 1.11100\ E\text{-}4$

$P_2(8,11) = q_5 P_6 = 4.16613\ E\text{-}5$

$P_2(8,12) = q_5 q_6 = 4.629\ E\text{-}9$

$P_2(9,9) = P_4 = .99998333$

$P_2(9,12) = q_4 = 1.66666\ E\text{-}5$

$P_2(10,10) = P_5 = .99995833$

$P_2(10,12) = q_5 = 4.16659\ E\text{-}5$

$P_2(11,11) = P_6 = .99988890$

$P_2(11,12) = q_6 = 1.11105\ E\text{-}4$

Summarizing the probabilities:

$$\begin{aligned} Pr(F_2=2) &= R_r(V_1) \\ &= .99877758 \\ Pr(F_2=1) &= P_r(V_2) + P_r(V_3) \\ &= 0 + 1.21553\ E\text{-}3 \\ &= 1.21553\ E\text{-}3 \\ Pr(F_2=0) &= P_r(V_4) + P_r(V_5) + P_r(V_6) \\ &= 6.5208\ E\text{-}6 + 3.4073\ E\text{-}7 + 2.0674\ E\text{-}8 \\ &= 6.8822\ E\text{-}6. \end{aligned}$$

Subspace Q3. Each $(\alpha_3 K_2)^{-1}(f_3)$, $f_3 \in 0,1,2$, is a union of Cartesian sets in $U_3^2 = Q_3 \times Q_3$ where $Q_3 = \left\{ \begin{bmatrix} x_7 \\ x_8 \\ x_9 \end{bmatrix} \right\}$.

From Table 1 we have:

	No Diversion		Safety
	Phase 1	Phase 2	Phases 1 and 2
x_7	2	1 or 2	1 or 2
x_8	2	1 or 2	1 or 2
x_9	2	1 or 2	1 or 2

Each $q = \begin{bmatrix} x_7 \\ x_8 \\ x_9 \end{bmatrix}$ is a state in Q_3. Figure 5 presents the state diagram for Q_3.

A state number is associated with each $q \in Q_3$. Using the state numbers, we can rewrite the above table as:

	No Diversion		Safe Flight
Condition	Phase 1	Phase 2	Phases 1 and 2
states	1	1-8	1-8

The $(\alpha_3 K_2)^{-1}(f_3)$ sets can now be written in terms of Cartesian sets of states in Q_3:

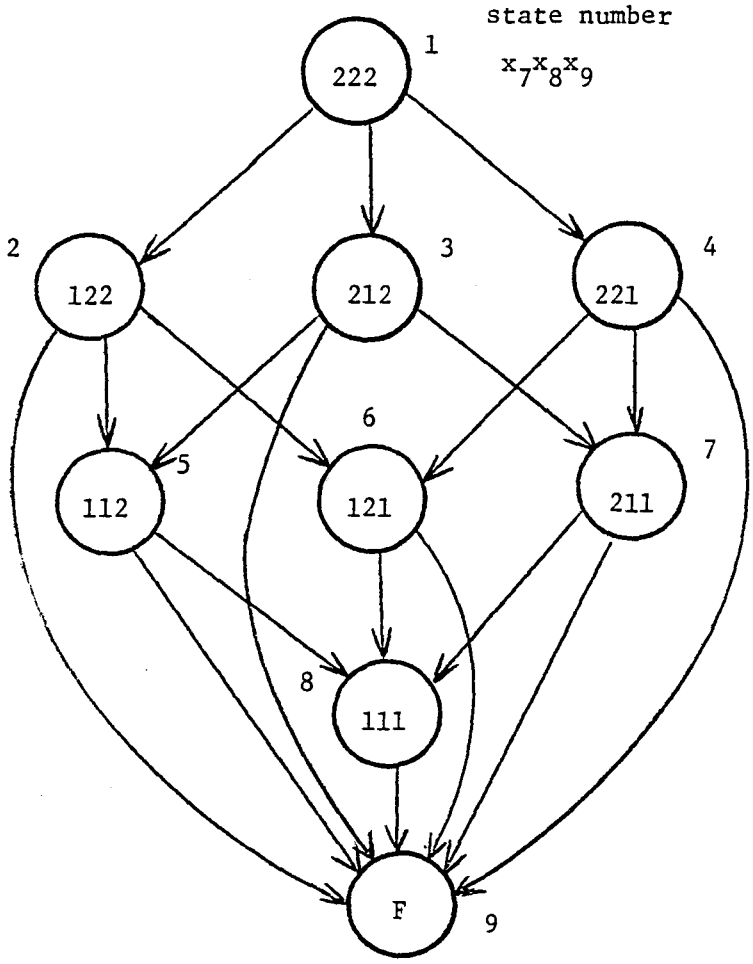


FIGURE 5. STATE DIAGRAM FOR S_3

$Q_3 = \{1,2,3,4,5,6,7,8,9\}$

$U_3^2 = Q_3 \times Q_3$

$(\alpha_3 K_2)^{-1}(2) = V_1$

where

V_1 = {no diversion, phase 1} x {no diversion, phase 2}

V_1 = {1} x {1,2,3,4,5,6,7,8}

$(\alpha_3 K_2)^{-1}(1) = V_2 \cup V_3$

where

V_2 = {no diversion, phase 1} x {diversion and safe, phase 2}

V_2 = {1} x ϕ

and

V_3 = {diversion, phase 1} x {safe flight, phase 2}

V_3 = {2,3,4,5,6,7,8} x {1,2,3,4,5,6,7,8}

$(\alpha_3 K_2)^{-1}(0) = V_4 \cup V_5$

where

V_4 = {unsafe, phase 1} x {all states, phase 2}

V_4 = {9} x Q

and

V_5 = {safe, phase 1} x {unsafe, phase 2}

V_5 = {1,2,3,4,5,6,7,8} x {9}

Figure 6 shows the intraphase transition matrix. The symbolic computations for $Pr(V_i)$ are as follows:

V_1 = {1} x {1,2,3,4,5,6,7,8}

$I(o) \cdot P_{v,1} \cdot G_{v1,1} = [P_1(1,1)\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$

$$P_{Vk} = [P_k(i,j)]$$

$$= \begin{bmatrix} P(1,1) & P(1,2) & P(1,3) & P(1,4) & P(1,5) & P(1,6) & P(1,7) & P(1,8) & P(1,9) \\ 0 & P(2,2) & 0 & 0 & P(2,5) & P(2,6) & 0 & P(2,8) & P(2,9) \\ 0 & 0 & P(3,3) & 0 & P(3,5) & 0 & P(3,7) & P(3,8) & P(3,9) \\ 0 & 0 & 0 & P(4,4) & 0 & P(4,6) & P(4,7) & P(4,8) & P(4,9) \\ 0 & 0 & 0 & 0 & P(5,5) & 0 & 0 & P(5,8) & P(5,9) \\ 0 & 0 & 0 & 0 & 0 & P(6,6) & 0 & P(6,8) & P(6,9) \\ 0 & 0 & 0 & 0 & 0 & 0 & P(7,7) & P(7,8) & P(7,9) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & P(8,8) & P(8,9) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

FIGURE 6. INTRAPHASE TRANSITION MATRIX FOR $V \leq Q_3$ AND PHASE K

$$P_{v,2} \cdot G_{v1,2} \cdot F$$

$$= \begin{bmatrix} P_2(1,1) + P_2(1,2) + P_2(1,3) + P_2(1,4) + P_2(1,5) + P_2(1,6) + P_2(1,7) + P_2(1,8) \\ P_2(2,2) + P_2(2,5) + P_2(2,6) + P_2(2,8) \\ P_2(3,3) + P_2(3,5) + P_2(3,7) + P_2(3,8) \\ P_2(4,4) + P_2(4,6) + P_2(4,7) + P_2(4,8) \\ P_2(5,5) + P_2(5,8) \\ P_2(6,6) + P_2(6,8) \\ P_2(7,7) + P_2(7,8) \\ P_2(8,8) \\ 0 \end{bmatrix}$$

$$Pr(V_1) = P_1(1,1) \cdot \sum_{j=1}^{8} P_2(1,j)$$

$$V_2 = \{1\} \times \phi$$

$$Pr(V_2) = 0 \quad \text{since } Pr(\phi) = 0$$

$$V_3 = \{2,3,4,5,6,7,8\} \times \{1,2,3,4,5,6,7,8\}$$

$$I(o) \cdot P_{v,1} \cdot G_{v3,1}$$

$$= [0 \; P_1(1,2) \; P_1(1,3) \; P_1(1,4) \; P_1(1,5) \; P_1(1,6) \; P_1(1,7) \; P_1(1,8) \; 0]$$

$$P_{v,2} \cdot G_{v3,2} \cdot F$$

$$= \begin{bmatrix} P_2(1,1) + P_2(1,2) + P_2(1,3) + P_2(1,4) + P_2(1,5) + P_2(1,6) + P_2(1,7) + P_2(1,8) \\ P_2(2,2) + P_2(2,5) + P_2(2,6) + P_2(2,8) \\ P_2(3,3) + P_2(3,5) + P_2(3,7) + P_2(3,8) \\ P_2(4,4) + P_2(4,6) + P_2(4,7) + P_2(4,8) \\ P_2(5,5) + P_2(5,8) \\ P_2(6,6) + P_2(6,8) \\ P_2(7,7) + P_2(7,8) \\ P_2(8,8) \\ 0 \end{bmatrix}$$

$$\begin{aligned} Pr(V_3) = {} & P_1(1,2)\ [P_2(2,2) + P_2(2,5) + P_2(2,6) + P_2(2,8)] \\ & + P_1(1,3)\ [P_2(3,3) + P_2(3,5) + P_2(3,7) + P_2(3,8)] \\ & + P_1(1,4)\ [P_2(4,4) + P_2(4,6) + P_2(4,7) + P_2(4,8)] \\ & + P_1(1,5)\ [P_2(5,5) + P_2(5,8)] \\ & + P_1(1,6)\ [P_2(6,6) + P_2(6,8)] \\ & + P_1(1,7)\ [P_2(7,7) + P_2(7,8)] \\ & + P_1(1,8) \cdot P_2(8,8) \end{aligned}$$

$$V_4 = \{9\} \times Q$$

$$I(0) \cdot P_{v,1} \cdot G_{v4,1} = [0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ P_1(1,9)]$$

$$P_{v,2} \cdot G_{v4,2} \cdot F = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^t$$

$$Pr(V_4) = P_1(1,9)$$

$$V_5 = \{1,2,3,4,5,6,7,8\} \quad x \ \{9\}$$

$$I(o) \cdot P_{v,1} \cdot G_{v5,1}$$

$$= [P_1(1,1)\ P_1(1,2)\ P_1(1,3)\ P_1(1,4)\ P_1(1,5)\ P_1(1,6)\ P_1(1,7)\ P_1(1,8)\ 0]$$

$$P_{v,2} \cdot G_{v5,2} \cdot F = \begin{bmatrix} P_2(1,9) \\ P_2(2,9) \\ P_2(3,9) \\ P_2(4,9) \\ P_2(5,9) \\ P_2(6,9) \\ P_2(7,9) \\ P_2(8,9) \\ P_2(9,9) \end{bmatrix}$$

$$Pr(V_5) = \sum_{J=1}^{8} P_1(1,j) \cdot P_2(j,9)$$

Using the MTBF data from the problem statement, $P_n = \exp(-\lambda_n t_k)$, and $q_n = 1-P_n$, we have:

	Phase 1		Phase 2	
n	P_n	q_n	P_n	q_n
7	.99756962	2.43038 E-3	.99993334	6.66645 E-5
8	.99939185	6.08148 E-4	.99998333	1.66666 E-5
9	.99756962	2.43038 E-3	.99993334	6.66645 E-5

The $P_k(i,j)$ computations for Q_3 are as follows:

$P_1(1,1) = P_7^2 P_8^2 P_9^2 = .989110$

$P_1(1,2) = 2P_7 q_7 P_8^2 P_9^2 = 4.81953 \text{ E-3}$

$P_1(1,3) = 2P_7^2 P_8 q_8 P_9^2 = 1.20378 \text{ E-3}$

$P_1(1,4) = 2P_7^2 P_8^2 P_9 q_9 = 4.81953 \text{ E-3}$

$P_1(1,5) = 4P_7 q_7 P_8 q_8 P_q^2 = 5.86554 \text{ E-6}$

$P_1(1,6) = 4P_7 q_7 P_8 2P_9 q_9 = 2.34836 \text{ E-5}$

$P_1(1,7) = 4P_7^2 P_8 q_8 P_9 q_9 = 5.86554 \text{ E-6}$

$P_1(1,8) = 8P_7 q_7 P_8 q_8 P_9 q_9 = 2.85804 \text{ E-8}$

$P_1(1,9) = q_7^2 + (1-q_7^2) q_8^2 + (1-q_7^2)(1-q_8^2) q_9^2 = 1.21833 \text{ E-5}$

$P_2(1,1) = .999700$

$P_2(1,2) = 1.33298 \text{ E-4}$

$P_2(1,3) = 3.33238 \text{ E-5}$

$P_2(1,4) = 1.33298 \text{ E-4}$

$P_2(1,5) = 4.44332 \text{ E-9}$

$P_2(1,6) = 1.77737 \text{ E-8}$

$P_2(1,7) = 4.44332 \text{ E-9}$

$P_2(1,8) = 5.9 \text{ E-13}$

$P_2(1,9) = 9.1661 \text{ E-9}$

$P_2(2,2) = P_7 P_8^2 P_9^2 = .9997667$

$P_2(2,5) = 2P_7 P_8 q_8 P_9^2 = 3.33260 \text{ E-5}$

$P_2(2,6) = 2P_7 P_8^2 P_9 q_9 = 1.33307 \text{ E-4}$

$P_2(2,8) = 4P_7 P_8 q_8 P_9 q_9 = 4.44362 \text{ E-9}$

$P_2(2,9) = q_7 + P_7 q_8^2 + P_7(1-q_8^2) q_9^2 = 6.66692 \text{ E-5}$

$$P_2(3,3) = P_7^2 P_8 P_9^2 = .999717$$

$$P_2(3,5) = 2P_7 q_7 P_8 P_9^2 = 1.33300\ E\text{-}4$$

$$P_2(3,7) = 2P_7^2 P_8 P_9 q_9 = 1.33300\ E\text{-}4$$

$$P_2(3,8) = 4P_7 q_7 P_8 P_9 q_9 = 1.7774\ E\text{-}8$$

$$P_2(3,9) = q_8 + P_8 q_7^2 + P_8(1-q_7^2) q_9^2 = 1.66755\ E\text{-}5$$

$$P_2(4,4) = P_7^2 P_8^2 P_9 = .9997667$$

$$P_2(4,6) = 2P_7 q_7 P_8^2 P_9 = 1.33307\ E\text{-}4$$

$$P_2(4,7) = 2P_7^2 P_8 q_8 P_9 = 3.33260\ E\text{-}5$$

$$P_2(4,8) = 4P_7 q_7 P_8 q_8 P_9 = 4.44362\ E\text{-}9$$

$$P_2(4,9) = q_9 + P_9 q_7^2 + P_9(1-q_7^2) q_8^2 = 6.66692\ E\text{-}5$$

$$P_2(5,5) = P_7 P_8 P_9^2 = .999783$$

$$P_2(5,8) = 2P_7 P_8 P_9 q_9 = 1.33309\ E\text{-}4$$

$$P_2(5,9) = q_7 + P_7 q_8 + P_7 P_8 q_9^2 = 8.33344\ E\text{-}5$$

$$P_2(6,6) = P_7 P_8^2 P_9 = .999833$$

$$P_2(6,8) = 2P_7 P_8 q_8 P_9 = 3.33282\ E\text{-}5$$

$$P_2(6,9) = q_7 + P_7 q_9 + P_7 P_9 q_8^2 = 1.33325\ E\text{-}4$$

$$P_2(7,7) = P_7^2 P_8 P_9 = .999783$$

$$P_2(7,8) = 2P_7 q_7 P_8 P_9 = 1.33309\ E\text{-}4$$

$$P_2(7,9) = q_8 + P_8 q_9 + P_8 P_9 q_7^2 = 8.33344\ E\text{-}5$$

$$P_2(8,8) = P_7 P_8 P_9 = .999850$$

$$P_2(8,9) = q_7 + P_7 q_8 + P_7 P_8 q_9 = .000150$$

Summarizing the probabilities:

$$\Pr(F_3 = 2) = \Pr(V_1)$$
$$= .989110$$
$$\Pr(F_3 = 1) = \Pr(V_2) + \Pr(V_3)$$
$$= 0 + .0108774$$
$$= .0108774$$
$$\Pr(F_3 = 0) = \Pr(V_4) + \Pr(V_5)$$
$$= 1.21833 \text{ E-5} + 6.75688 \text{ E-7}$$
$$= 1.2859 \text{ E-5}.$$

Subspace Q_4. Each $(\alpha_4 K_2)^{-1}(f_4)$, $f_4 \in \{0,1,2\}$, is a union of Cartesian sets in $U_4^2 = Q_4 \times Q_4$ where

$$Q_4 = \left\{ \begin{bmatrix} x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \end{bmatrix} \right\}$$

Since x_{10}, $x_{12} \in \{0,1\}$ and x_{11}, $x_{13} \in \{0,1,2\}$, the number of states in Q_4 is $2 \cdot 3 \cdot 2 \cdot 3 = 36$. Lumping all states which correspond to unsafe flight reduces the number of states to 17. To reduce this number to a more manageable value, we will introduce an additional model level which describes processing units (PU). A PU is defined to be a FCC and its associated BIU's. We will first model the behavior of the function f_4 in terms of PU's, and then model each PU in terms of its components. From the component model we will then be combined to derive the probabilities for the function f_4.

Let Y_i be the random variable which denotes the state of PU_i ($i = 1,2$) where the states are defined as follows:

FCC - i (x_{10} or x_{12})	BIU - i (x_{11} or x_{13})	Y_i
1	2	2
1	1	1
1	0	0
0	*	0

where * represents "any possible value".

Let $\hat{Q}_4 = \left\{ \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \mid y_i \in \{0,1,2\} \right\}$. The corresponding trajectory space is $\hat{U}_4 = \hat{Q}_4 \times \hat{Q}_4$. Based on Table 1, the Y_i values for no diversion and safety can be specified as follows:

	No Diversion		Safe Flight
	Phase 1	Phase 2	Phases 1 and 2
Y_1	2	(1 or 2) (0)	(1 or 2) (0)
Y_2	2	(*) (1 or 2)	(*) (1 or 2)

Let L denote the mapping $L : \hat{U}_4 \longrightarrow \{f_4\}$. Using the above table and the state diagram for $\hat{Q}_4$ in Figure 7, we can specify the inverses $L^{-1}(f_4)$ in terms of the state numbers:

$L^{-1}(2) = V_1$

where

V_1 = {no diversion, phase 1} x {no diversion, phase 2}

$V_1 = \{1\} \times \{1,2,3,4,5,6,7,8\}$

$L^{-1}(1) = V_2 \cup V_3$

where

V_2 = {no diversion, phase 1} x {diversion and safe, phase 2}

$V_2 = 1 \times \phi$

and

V_3 = {diversion and safe, phase 1} x {safe, phase 2}

$V_3 = \{2,3,4,5,6,7,8\} \times \{1,2,3,4,5,6,7,8\}$

$L^{-1}(0) = V_4 \cup V_5$

where

V_4 = {unsafe, phase 1} x {all states, phase 2}

$V_4 = 9 \times \hat{Q}_4$

and

V_5 = {safe, phase 1} x {unsafe, phase 2}

$V_5 = \{1,2,3,4,5,6,7,8\} \times \{9\}$.

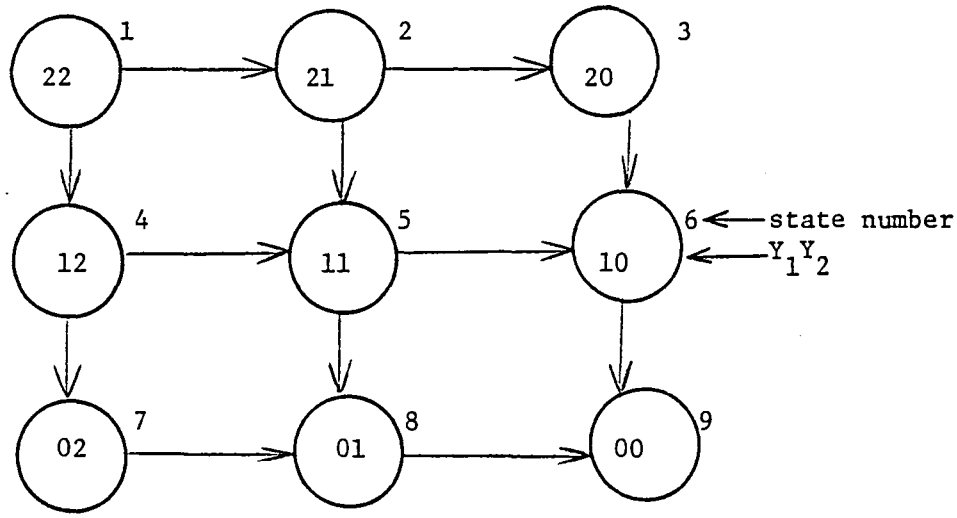


FIGURE 7. STATE DIAGRAM FOR $\hat{Q}_4$ AND EACH PHASE

The intraphase transition matrix $P_{v,k}$ is shown in Figure 8. The symbolic computations for $Pr(V_i)$ are as follows:

$$V_1 = \{1\} \times \{1,2,3,4,5,6,7,8\}$$

$$G_{V_1,1} = \begin{bmatrix} 1 & & & & & & & \\ & 0 & & & & & & \\ & & 0 & & & & & \\ & & & 0 & & & & \\ & & & & 0 & & & \\ & & & & & 0 & & \\ & & & & & & 0 & \\ & & & & & & & 0 \end{bmatrix} \qquad G_{V_1,2} = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 0 \end{bmatrix}$$

$$I(0) = [1\ 0\ 0\ 0\ 0\ 0\ 0\ 0]$$

$$F = [1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]^t$$

$$Pr(V_1) = I(0) \cdot P_{V,1} \cdot G_{V_1,1} \cdot P_{V,2} \cdot G_{V_1,2} \cdot F$$

$$Pr(V_1) = P_1(1,1) \cdot \sum_{j=1}^{8} P_2(1,j)$$

$$V_2 = \{1\} \times \phi$$

$Pr(V_2) = 0$ since $Pr(\phi) = 0$.

$$V_3 = \{2,3,4,5,6,7,8\} \times \{1,2,3,4,5,6,7,8\}$$

$$G_{V3,1} = \begin{bmatrix} 0 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 0 \end{bmatrix} \qquad G_{V3,2} = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 0 \end{bmatrix}$$

I(0) and F are as above.

$$Pr(V_3) = \sum_{j=2}^{8} \left(P_1(1,j) \cdot \sum_{k=1}^{8} P_2(j,k) \right)$$

$$P_{V,K} = \begin{bmatrix} P(1,1) & P(1,2) & P(1,3) & P(1,4) & P(1,5) & P(1,6) & P(1,7) & P(1,8) & P(1,9) \\ 0 & P(2,2) & P(2,3) & 0 & P(2,5) & P(2,6) & 0 & P(2,8) & P(2,9) \\ 0 & 0 & P(3,3) & 0 & 0 & P(3,6) & 0 & 0 & P(3,9) \\ 0 & 0 & 0 & P(4,4) & P(4,5) & P(4,6) & P(4,7) & P(4,8) & P(4,9) \\ 0 & 0 & 0 & 0 & P(5,5) & P(5,6) & 0 & P(5,8) & P(5,9) \\ 0 & 0 & 0 & 0 & 0 & P(6,6) & 0 & 0 & P(6,9) \\ 0 & 0 & 0 & 0 & 0 & 0 & P(7,7) & P(7,8) & P(7,9) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & P(8,8) & P(8,9) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

FIGURE 8. INTRAPHASE TRANSITION MATRIX FOR $\hat{Q}_4$ AND PHASE K. THE PHASE SUBSCRIPT K IS OMITTED FROM THE P(i,j) FOR CONVENIENCE

$$V_4 = \{9\} \times \hat{Q}_4$$

$\Pr(V_4) = P_1(1,9)$ since $\Pr(\hat{Q}_4) = 1$.

$$V_5 = \{1,2,3,4,5,6,7,8\} \times \{9\}$$

$G_{V5,1} =$

$$\begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & 1 & & & & & \\ & & & 1 & & & & \\ & & & & 1 & & & \\ & & & & & 1 & & \\ & & & & & & 1 & \\ & & & & & & & 0 \end{bmatrix}$$

$G_{V5,2} =$

$$\begin{bmatrix} 0 & & & & & & & \\ & 0 & & & & & & \\ & & 0 & & & & & \\ & & & 0 & & & & \\ & & & & 0 & & & \\ & & & & & 0 & & \\ & & & & & & 0 & \\ & & & & & & & 1 \end{bmatrix}$$

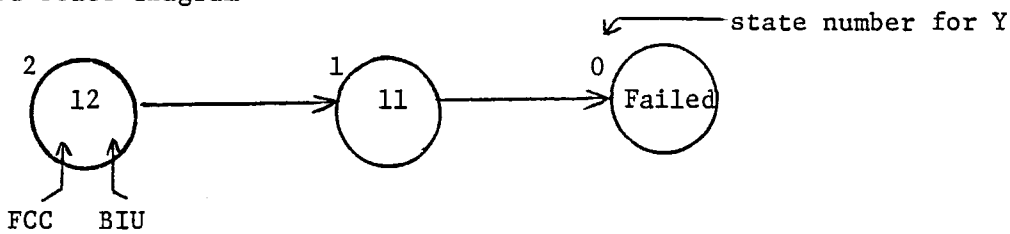
I(0) and F are as above.

$$\Pr(V_5) = \sum_{J=1}^{8} P_1(1,j) \cdot P_2(j,9).$$

Next, we can compute the $P_K(i,j)$ probabilities using the individual transition probabilities for each Y_i. Let

$$W_k(m,n) = \Pr[Y_i \text{ ends phase k in state n} \mid y_i \text{ begins phase k in state m}].$$

The associated state diagram is:



Let $P_F = \Pr[\text{FCC remains fault-free for the phase}]$ $q_F = 1-P_F$

$P_B = \Pr[\text{a BIU remains fault-free for the phase}]$ $q_B = 1-P_B$

Then the $W_k(m,n)$ may be expressed as follows:

$W_k(m,n)$		n = 2	n = 1	n = 0
m	2	$P_F P_B^2$	$2P_F P_B q_B$	$q_F + P_F q_B^2$
	1	0	$P_F P_B$	$q_F + P_F qB$
	0	0	0	1

Using MTBF data and phase durations,

	Phase 1	Phase 2
P_F	.99756962	.99993334
q_F	2.43038 E-3	6.66645 E-5
P_B	.99878407	.99996667
q_B	1.21593 E-3	3.33328 E-5

The above data are used to compute the $W_k(m,n)$:

$W_1(m,n)$		n = 2	n = 1	n = 0
m	2	.99514514	2.42299 E-3	2.43185 E-3
	1	0	.99635664	3.64335 E-3
	0	0	0	1

$W_2(m,n)$		n = 2	n = 1	n = 0
m	2	.99986668	6.66589 E-5	6.66656 E-5
	1	0	.99990001	9.99951 E-5
	0	0	0	1

Next, the $W_k(m,n)$ and the state diagram for $\hat{Q}_4$ (Figure 7) are used to compute the $P_k(i,j)$. Only those $P_k(i,j)$ with positive values are computed.

$P_1(1,1) = W_1(2,2)^2 = .990314$

$P_1(1,2) = W_1(2,2) \cdot W_1(2,1) = 2.41123\ E\text{-}3$

$P_1(1,3) = W_1(2,2) \cdot W_1(2,0) = 2.42004\ E\text{-}3$

$P_1(1,4) = W_1(2,1) \cdot W_1(2,2) = 2.41123\ E\text{-}3$

$P_1(1,5) = W_1(2,1)^2 = 5.87088\ E\text{-}6$

$P_1(1,6) = W_1(2,1) \cdot W_1(2,0) = 5.89235\ E\text{-}6$

$P_1(1,7) = W_1(2,0) \cdot W_1(2,2) = 2.42004\ E\text{-}3$

$P_1(1,8) = W_1(2,0) \cdot W_1(2,1) = 5.89235\ E\text{-}6$

$P_1(1,9) = W_1(2,0)^2 = 5.91389\ E\text{-}6$

$P_2(1,1) = .9997334$

$P_2(1,2) = 6.66500\ E\text{-}5$

$P_2(1,3) = 6.66577\ E\text{-}5$

$P_2(1,4) = 6.66500\ E\text{-}5$

$P_2(1,5) = 4.44341\ E\text{-}9$

$P_2(1,6) = 4.44392\ E\text{-}9$

$P_2(1,7) = 6.66577\ E\text{-}5$

$P_2(1,8) = 4.44392\ E\text{-}9$

$P_2(1,9) = 4.44443\ E\text{-}9$

$P_2(2,2) = W_2(2,2) \cdot W_2(1,1) = .999767$

$P_2(2,3) = W_2(2,2) \cdot W_2(1,0) = 9.99818\ E\text{-}5$

$P_2(2,5) = W_2(2,1) \cdot W_2(1,1) = 6.66522\ E\text{-}5$

$P_2(2,6) = W_2(2,1) \cdot W_2(1,0) = 6.66556\ E\text{-}9$

$P_2(2,8) = W_2(2,0) \cdot W_2(1,1) = 6.66599\ E\text{-}5$

$P_2(2,9) = W_2(2,0) \cdot W_2(1,0) = 6.66633\ E\text{-}9$

$P_2(3,3) = W_2(2,2) = .999867$

$P_2(3,6) = W_2(2,1) = 6.66589\ E\text{-}5$

$P_2(3,9) = W_2(2,0) = 6.66656\ E\text{-}5$

$P_2(4,4) = W_2(1,1) \cdot W_2(2,2) = .999767$

$P_2(4,5) = W_2(1,1) \cdot W_2(2,1) = 6.66522\ E\text{-}5$

$P_2(4,6) = W_2(1,1) \cdot W_2(2,0) = 6.66599\ E\text{-}5$

$P_2(4,7) = W_2(1,0) \cdot W_2(2,2) = 9.99818\ E\text{-}5$

$P_2(4,8) = W_2(1,0) \cdot W_2(2,1) = 6.66556\ E\text{-}9$

$P_2(4,9) = W_2(1,0) \cdot W_2(2,0) = 6.66633\ E\text{-}9$

$P_2(5,5) = W_2(1,1)^2 = .999800$

$P_2(5,6) = W_2(1,1) \cdot W_2(1,0) = 9.99851\ E\text{-}5$

$P_2(5,8) = W_2(1,0) \cdot W_2(1,1) = 9.99851\ E\text{-}5$

$P_2(5,9) = W_2(1,0)^2 = 9.99902\ E\text{-}9$

$P_2(6,6) = W_2(1,1) = .999900$

$P_2(6,9) = W_2(1,0) = 9.99951\ E\text{-}5$

$P_2(7,7) = W_2(2,2) = .999867$

$P_2(7,8) = W_2(2,1) = 6.66589\ E\text{-}5$

$P_2(7,9) = W_2(2,0) = 6.66656\ E\text{-}5$

$P_2(8,8) = W_2(1,1) = .999900$

$P_2(8,9) = W_2(1,0) = 9.99951\ E\text{-}5$

$P_2(9,9) = 1.00$

The $P_K(i,j)$ are used to compute the Pr(Vi) according to the equations previously derived. The results are:

$Pr(V_1) = .990314$

$Pr(V_2) = 0$

$Pr(V_3) = 9.67988\ E\text{-}3$

$Pr(V_4) = 5.91389\ E\text{-}6$

$Pr(V_5) = 3.2827939$

Finally, the probabilities for the function f_4 are computed:

$$Pr(F_4 = 2) = Pr(L^{-1}(2)) = Pr(V_1)$$
$$= .990314$$
$$Pr(F_4 = 1) = Pr(L^{-1}(1)) = Pr(V_2) + Pr(V_3)$$
$$= 9.67987\ E\text{-}3$$
$$Pr(F_4 = 0) = Pr(L^{-1}(0)) = Pr(V_4) + Pr(V_5)$$
$$= 6.242174\ E\text{-}6$$

Final Computations. The preceding four subsections show the derivations of $Pr(F_i = f_i)$, $f_i \in \{0,1,2\}$ for each of the four Q_i subspaces. In this subsection, the preceding results are combined to compute $P(a_n)$, the probability of accomplishment level a_n, for each $a_n \in A$. The remaining steps for each a_n are as follows:

- For each $V = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \in \gamma_1^{-1}(a_n)$, compute

$$Pr(V) = \prod_{i=1}^{4} Pr(F_i = f_i)$$

- Sum the Pr(V) quantities for all $V \in \gamma_1^{-1}(a_n)$:

$$P(a_n) = Pr(\gamma_1^{-1}(a_n)) = \sum_{v \in \gamma_1^{-1}(a_n)} Pr(V).$$

The computation in the first step is based on the independence of the Q_i subspaces. The equation in the second step uses the fact that the elements V of $\gamma_1^{-1}(a_n)$ are mutually exclusive. The $Pr(F_i=f_i)$ values from the preceding analysis are presented in Table 3.

TABLE 3. CONTRIBUTIONS OF SUBSPACES TO MISSION OUTCOMES: $Pr(F_i = f_i)$ where $i = 1,2,3,4$ and $f_i \in \{0,1,2\}$.

i	$Pr(F_i = 2)$	$Pr(F_i = 1)$	$Pr(F_i = 0)$
1	.995788	4.21164 E-3	0
2	.998778	1.21553 E-3	6.8822 E-6
3	.989110	1.08774 E-2	1.28592 E-5
4	.990314	9.67987 E-3	6.24217 E-6

From the Level 1 Model discussion, $\gamma_1^{-1}(a_o) = \left\{\begin{bmatrix}2\\2\\2\\2\end{bmatrix}\right\}$.

$$\Pr\left(\begin{bmatrix}2\\2\\2\\2\end{bmatrix}\right) = \prod_{i=1}^{4} \Pr(F_i=2) = .974212.$$

Hence, $P(a_o) = .974212$.

Next, $\gamma_1^{-1}(a_1) = \{v_1, v_2, v_3, v_4\}$ where

$$v_1 = \begin{bmatrix}1\\1 \text{ or } 2\\1 \text{ or } 2\\1 \text{ or } 2\end{bmatrix}, \quad v_2 = \begin{bmatrix}2\\1\\1 \text{ or } 2\\1 \text{ or } 2\end{bmatrix}, \quad v_3 = \begin{bmatrix}2\\2\\1\\1 \text{ or } 2\end{bmatrix}, \quad v_4 = \begin{bmatrix}2\\2\\2\\1\end{bmatrix}.$$

$$\Pr(v_1) = \Pr(F_1=1) \cdot \big(\Pr(F_2=1) + \Pr(F_2=2)\big) \cdot \big(\Pr(F_3=1) + \Pr(F_3=2)\big) \cdot \big(\Pr(F_4=1) + \Pr(F_4=2)\big)$$

$$= 4.21153 \text{ E-3}$$

Similarly,

$\Pr(v_2) = 1.21039$ E-3

$\Pr(v_3) = 1.08183$ E-2

$\Pr(V_4) = 9.52248$ E-3

Summing the $\Pr(V_j)$ yields $P(a_1) = .025763$.

Next, $\gamma^{-1}(a_2) = \{v_5, v_6, v_7, v_8\}$ where

$$v_5 = \begin{bmatrix}0\\*\\*\\*\end{bmatrix}, \quad v_6 = \begin{bmatrix}1 \text{ or } 2\\0\\*\\*\end{bmatrix}, \quad v_7 = \begin{bmatrix}1 \text{ or } 2\\1 \text{ or } 2\\0\\*\end{bmatrix}, \quad v_8 = \begin{bmatrix}1 \text{ or } 2\\1 \text{ or } 2\\1 \text{ or } 2\\0\end{bmatrix}.$$

Since * represents "any feasible value", $\Pr(F_i = *) = 1.0$.

To insure numerical accuracy, the values $\Pr(F_i = 0)$ are computed as follows:

$$\Pr(F_i = 0) = 1 - \big(\Pr(F_i = 1) + \Pr(F_i = 2)\big).$$

Using the probability values from Table 3,

$\Pr(v_5) = \Pr(F_1 = 0) = 0$

$\Pr(v_6) = \big[\Pr(F_1=1) + \Pr(F_1=2)\big] \cdot \Pr(F_2=0) = 6.88218$ E-6

$\Pr(v_7) = \big[\Pr(F_1=1) + \Pr(F_1=2)\big] \big[\Pr(F_2=1) + \Pr(F_2=2)\big] \cdot \Pr(F_3=0)$

$= 1.285904$ E-5

$$\Pr(v_8) = \left[\Pr(F_1=1) + \Pr(F_1=2)\right] \cdot \left[\Pr(F_2=1) + \Pr(F_2=2)\right]$$
$$\cdot \left[\Pr(F_3=1) + \Pr(F_3=2)\right] \cdot \Pr(F_4=0)$$
$$= 6.24205 \text{ E-6}$$

Summing the above $\Pr(v_j)$ yields $P(a_2) = 28.9833$ E-6.

In summary, the mission outcome probabilities are:

Safe, no diversion: $P(a_o) = .974212$

Safe, diversion : $P(a_1) = .025763$

Unsafe : $P(a_2) = 25.9833 \times 10^{-6}$.

Dependencies Not Captured

As noted in the discussion of the application of performability analysis to the dual-dual problem (see the section entitled "Analysis Results"), an error was made in selecting groups of components which were independent with respect to their impacts on the mission outcome. The following paragraphs explain the dependencies in question and provide an upper bound for their probability of occurrence.

The performability analysis solution treated the components in independent sets. In particular, components 1 and 2 were in one set (call it C_1) while components 3, 4, and 5 comprised a different set (call it C_2). The probabilities of set C_1 resulting in mission accomplishment level a_j (a_0=no diversion, safe; a_1=diversion, safe; a_2=unsafe) were computed independent of the state of the set C_2. Similarly, the probabilities of C_2 resulting in a_j were computed independent of C_1. However, C_1 and C_2 are not independent.

There are two cases (i.e., mission profiles) in which C_1 and C_2 must be considered simultaneously to determine the correct mission outcome. In each case, the CAT II landing is initiated, after which both DAD's fail. In this state, the CAT II landing can still be completed (even though the safe flight conditions are not satisfied). A subsequent failure of the radar altimeter or of both VOR's causes violation of the conditions required to complete the CAT II landing, thereby causing a diversion. When a diversion occurs, the safe flight conditions must be met or the aircraft is lost. Since both DAD's are failed, the aircraft is lost. When the sets C_1 and C_2 were treated independently, both of the above cases were treated as if the mission

outcome was diversion and safe flight. Note that if either the radar altimeter or both VOR's fail first, (i.e., prior to failure of both DAD's) then the CAT II landing is aborted and the aircraft is lost. These possibilities were captured correctly by the analysis.

To compute an upper bound for the probability of occurrence of the two above cases, let E_1 represent the two cases; i.e.,

E_1=the event the CAT II landing is initiated; and both DAD's fail before the landing is completed; and then either the radar altimeter or the second VOR fails before the landing is completed.

Let

E_2=the event the CAT II landing is initiated, and both DAD's and either the radar altimeter or both VOR's fail before the landing is completed.

Clearly, $Pr(E_1) \leq Pr(E_2)$.

Next, let

E_3=the event the CAT II landing is initiated.

According to the well-known Baye's Theorem,

$$Pr(E_2)=Pr(E_2|E_3)\cdot Pr(E_3).$$

Since $Pr(E_3) \leq 1$, then $Pr(E_2) \leq Pr(E_2|E_3)$.

Combining this inequality with $Pr(E_1) \leq Pr(E_2)$ implies

$$Pr(E_1) \leq Pr(E_2|E_3).$$

Hence, an upper bound for the event of interest, E_1, is the probability both DAD's fail in a two minute period and either the radar altimeter or

both VOR's fail in a two minute period. Using the MTBF data from Table 1 of the dual-dual problem statement,

$$\Pr(E_2|E_3) \leq \left(1-e^{-\frac{1}{2000}\cdot\frac{2}{60}}\right)^2 \cdot \left[\left(1-e^{-\frac{1}{700}\cdot\frac{2}{60}}\right) + \left(1-e^{-\frac{1}{1000}\cdot\frac{2}{60}}\right)^2\right]$$

$$\Pr(E_2|E_3) \leq 1.33 \times 10^{-14}$$

and therefore

$$\Pr(E_1) \leq 1.33 \times 10^{-14}.$$

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA CR-159358		

4. Title and Subtitle	5. Report Date
COMPARATIVE ANALYSIS OF TECHNIQUES FOR EVALUATING THE EFFECTIVENESS OF AIRCRAFT COMPUTING SYSTEMS	April 1981
	6. Performing Organization Code

7. Author(s)	8. Performing Organization Report No.
E. F. Hitt, M. S. Bridgman, and A. C. Robinson	
	10. Work Unit No.

9. Performing Organization Name and Address	11. Contract or Grant No.
Battelle Columbus Laboratories 505 King Avenue Columbus, Ohio 43201	NAS1-15760
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Langley Research Center Hampton, Virginia 23665	Contractor Report
	14. Sponsoring Agency Code

15. Supplementary Notes

Langley technical monitor: G. E. Migneault

16. Abstract

Performability analysis is a technique developed under NASA Grant NSG 1306 for evaluating the effectiveness of fault-tolerant computing systems in multi-phase missions. In this study, performability was evaluated for its accuracy, practical usefulness, and relative cost. The evaluation was performed by applying performability and the fault tree method to a set of sample problems ranging from simple to moderately complex. The problems involved as many as five outcomes, two to five mission phases, permanent faults, and some functional dependencies. Transient faults and software errors were not considered. A different analyst was responsible for each technique. This report describes the sample problems and their solutions using each method.

Significantly more time and effort were required to learn performability analysis than the fault tree method. Performability is inherently as accurate as fault tree analysis. For the sample problems, fault trees were more practical and less time-consuming to apply, while performability required less ingenuity and was more "checkable". Performability may offer some advantages for evaluating very complex problems.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Reliability analysis Fault-tolerant computing Performability	

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	147	

End of Document