

Archive

QUALITATIVE AND QUANTITATIVE RELIABILITY
ANALYSIS OF SAFETY SYSTEMS

by

R. Karimi, N. Rasmussen and L. Wolf

Energy Laboratory Report No. MIT-EL 80-015

May 1980

QUALITATIVE AND QUANTITATIVE RELIABILITY
ANALYSIS OF SAFETY SYSTEMS

by

R. Karimi, N. Rasmussen and L. Wolf

Energy Laboratory
and
Department of Nuclear Engineering

Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

sponsored by
Boston Edison Company
under the
MIT Energy Laboratory Electric Utility Program

Energy Laboratory Report No. MIT-EL 80-015

May 1980

QUALITATIVE & QUANTITATIVE RELIABILITY
ANALYSIS OF SAFETY SYSTEMS

by

Roohollah Karimi

Submitted to the Department of Nuclear Engineering on May, 1980
in partial fulfillment of the requirements for the degree of
Doctor of Science.

ABSTRACT

A code has been developed for the comprehensive analysis of a fault tree. The code designated UNRAC (UNReliability Analysis Code) calculates the following characteristics of an Input fault tree:

- a) minimal cut sets,
- b) top event unavailability as point estimate and/or in time dependent form,
- c) quantitative importance of each component involved, and
- d) error bound on the top event unavailability

UNRAC can analyze fault trees, with any kind of gates (EOR, NAND, NOR, AND, OR), up to a maximum of 250 components and/or gates.

For generating minimal cut sets the method of bit manipulation is employed. In order to calculate each component's time dependent unavailability, a general and consistent set of mathematical models is developed and the repair time density function is allowed to be represented by constant, exponential, 2nd order erlangian and log-normal distributions. A normally operating component is represented by a three-state model in order to be able to incorporate probabilities for revealed faults, non-revealed faults and false failures in unavailability calculations.

For importance analysis, a routine is developed that will rearrange the fault tree to evaluate the importance of each component to system failure, given that a component and/or a sub-system is unavailable (ie. down or failed). The importance of each component can be evaluated based on the instantaneous or average unavailabilities of each components. To simulate the distribution of top event uncertainty, a Monte-Carlo sampling routine is used. This method allows the user to input uncertainties on the component's failure characteristics (ie. failure rate, average test time, average repair

time, etc.) and assign different distributions for subsequent simulation.

The code is benchmarked against WAMCUT, MODCUT, KITT, BIT-FRANTIC and PL-MODT. The results showed that UNRAC produces results more consistent with the KITT results than either BIT-FRANTIC or PL-MODT. Overall it is demonstrated that UNRAC is an efficient and easy to use code and has the advantage of being able to do a complete fault tree analysis with this single code.

Thesis Advisor: Norman C. Rasmussen
Title: Head of the Department of Nuclear
Engineering

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	2
ACKNOWLEDGEMENTS.....	4
TABLE OF CONTENTS.....	5
LIST OF FIGURES	9
LIST OF TABLES	13
CHAPTER 1: INTRODUCTION	16
1.1 Reliability Concept and Methods of Evaluations	16
1.2 Fault Tree and Reliability Analysis	19
1.3 Fault Tree Evaluation	22
1.3.1 Qualitative Analysis	22
1.3.2 Quantitative Analysis	23
1.4 Objective of Thesis	26
1.5 Structure of Thesis	27
1.6 Glossary of Words	28
CHAPTER 2: ON THE ESSENTIAL OF RELIABILITY & PROB- ABILITY THEORY	33
2.1 Introduction and Basic Concept	33
2.2 Reliability Model of Non-Maintained Systems	38
2.3 Reliability Model of Maintained Systems	41
2.3.1 Introduction	41
2.3.2 Semi-Markovian Model	48

TABLE OF CONTENTS (Cont.)

	<u>Page</u>
2.3.3 Flow Graph and Mason's Rule ...	55
2.4 Reliability Model of Periodically Maintained Systems	63
CHAPTER 3: PRINCIPLE STRUCTURE OF THE CODE UNRAC ..	80
3.1 Introduction	80
3.2 Cut Set Generator.....	82
3.2.1 Introduction	82
3.2.2 Cut Set Generator Used In UNRAC	84
3.3 Unavailability Evaluator	98
3.3.1 Introduction	98
3.3.2 Unavailability Evaluator Used In UNRAC	104
3.4 Uncertainty Bounds Evaluator	117
3.4.1 Introduction	117
3.4.2 Monte-Carlo Simulator Used In UNRAC	122
3.5 On The Distribution of Input Variables	128
3.6 Moment Matching Method	130
CHAPTER 4: APPLICATION AND RESULTS	137
4.1 Introduction	137
4.2 Auxiliary Feed Water System (AFWS), A Comparison With WASH 1400	137

TABLE OF CONTENTS (Cont.)

	<u>Page</u>
4.3 An Example of An Electrical System, Comparison of UNRAC with FRANTIC and BIT-FRANTIC	164
4.4 A Chemical and Volume Control System (CVCS)	170
4.5 A High Pressure Injection System (HPIS), Comparison of WASH 1400 and UNRAC	
CHAPTER 5: SUMMARY, CONCLUSIONS AND RECOMMENDATIONS	187
5.1 Summary and Conclusions	187
5.2 Recommendations	192
REFERENCES	194
APPENDIX A: DETAILED MATHEMATICAL EXPRESSIONS OF EQUATIONS (2.39) AND (3.1)	203
A.1 Detailed Mathematical Form of Eqn. (2.39) for Exponential, 2nd Order Erlangian and Log-Normal	204
A.1.1 Exponential Repair Time Density Function	204
A.1.2 2nd Order Erlangian Repair Time Density Function	205
A.1.3 Log-Normal Repair Time Density Function	206
A.2 Detailed Mathematical Form of the Eqn. (3.1) for Different Repair Density Functions	208
A.2.1 Exponential Repair Time Density Function	208

TABLE OF CONTENTS (Cont.)

	<u>Page</u>
A.2.2 2nd Order Erlangian Repair Time Function	211
A.2.3 Log-Normal Repair Time Density Function	216
A.3 Solution of Equations of the Form $x = f(x)$	225
APPENDIX B: SOME BASIC BACKGROUND AND RELATED EXAMPLES USED IN SECTION 3.1	228
B.1 Mathematical Background for Boolean Operation	229
B.2 Application of UNRAC to the Two Fault Tree Examples, for Comparison with MODCUT and WAM-CUT	232
B.3 On the Evaluation of the Equivalent Component's Parameters	238
B.4 References	240
APPENDIX C: ON THE CODE STRUCTURE AND INPUT DESCRIPTION TO UNRAC	241
C.1 Code Structure	242
C.2 On the Random Number Generator and Sorting Routine Used in the Monte- Carlo Simulator MCSIM	246
C.3 INPUT Description of UNRAC	250
C.4 References	267
C.5 On the Input Sample Fault Tree	268

LIST OF FIGURES

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1.1	Overall reliability concept	17
1.2	Typical structuring of a fault tree ...	21
2.1	Typical failure rate behavior	40
2.2	A simple two-state transition diagram .	46
2.3	A typical transition diagram	49
2.4	Flow graph of equation (2.27)	56
2.5	Flow graph of equation (2.27)	56
2.6	Flow graph representation for the continuous time Semi-Markovian process ...	62
2.7	A typical time dependent unreliability of a periodically tested component	64
2.8	Comparison of the unavailabilities as calculated from equations developed in this study and those of Vesely's and Caldarola's for a periodically tested component	75
2.9	Comparison of the application of Eqn. (2.42) to different repair distributions	79
3.1	Flow chart and steps used in cut set generator BIT	86
3.2	Computer time to deterministically find minimal cut sets	88
3.3	Equivalent transformation of EOR, AND, NOR and NOT gates	89
3.4	Detailed information of operation and failure modes of an electrical wiring system	91

LIST OF FIGURES (Cont.)

<u>Figure</u>	<u>Title</u>	<u>Page</u>
3.5	Compact fault tree of the electrical wiring system	92
3.6	Fault tree showing a dependence of B on the occurrence of A	100
3.7	Dependent event connecting fault tree .	102
3.8	All possible path of a pump failure ...	105
3.9	A general three-state component	107
3.10	Comparison of cumulative probability distribution of a log-normal with its approximated combination of exponentials (median of 21 (hrs) and error factor of 2.56)	111
3.11	Comparison of cumulative probability distribution of a log-normal with its approximated combination of exponentials (median of 21 (hrs) and error factor of 1.4)	112
3.12	Comparison of cumulative probability distribution of a log-normal with its approximated combination of exponentials (median of 100 (hrs) and error factor of 3)	113
3.13	Logical flow chart used in UNRAC	116
3.14	Fault tree diagram minimal cut sets and failure data of an RPS example	118
3.15	A comparison of UNRAC with PREP & KITT, PL-MODT and BIT-FRANTIC	119
3.16	The execution time of the LIMITS and the SAMPLE codes	125
3.17	Region in (β_1, β_2) plane for various distributions	135

LIST OF FIGURES (Cont.)

<u>Figure</u>	<u>Title</u>	<u>Page</u>
3.18	Chart for determining appropriate Johnson distribution approximation	136
4.1	Simplified diagram of an AFWS	139
4.2	Simplified fault tree diagram of an AFWS	140
4.3	WASH 1400 qualitative results of an AFWS pictorial summary	142
4.4	A further reduced fault tree of AFWS input to UNRAC	143
4.5	AFWS time dependent unavailabilities as calculated by UNRAC	146
4.6	Cumulative top event distribution of AFWS, evaluated based on the average unavailability of each component	154
4.7	Cumulative top event distribution of AFWS, evaluated based on the individual component's failure characteristics ...	157
4.8	An example of an electrical system	166
4.9	A fault tree diagram of the example electrical system	167
4.10	Comparison of time dependent unavailabilities of the electrical system as calculated by UNRAC, FRANTIC and BIT-FRANTIC	169
4.11	A CVCS pump trains schematic diagram ..	171
4.12	A simplified fault tree diagram of CVCS pump trains	172

LIST OF FIGURES (Cont.)

<u>Figure</u>	<u>Title</u>	<u>Page</u>
4.13	CVCS time dependent unavailabilities as calculated by UNRAC and BIT- FRANTIC	176
4.14	A simplified HPIS diagram	180
4.15	A simplified fault tree diagram of the HPIS	181
B.1	A general Venn diagram	230
B.2	An example fault tree used to compare the results of BIT and MODCUT	233
B.3	An example fault tree used to compare the results of BIT and WAM-CUT	236
C.1	The general routines used in UNRAC	243
C.2	Exponential distribution random number generator flow diagram	247
C.3	Normal distribution random number generator flow diagram	248
C.4	Flow chart of sorting routine used in MCSIM	249

LIST OF TABLES

<u>Table</u>	<u>Title</u>	<u>Page</u>
2.1	Comparison of model developed in this study for periodically maintained component with Vesely's and Caldarola's model for a typical example	76
2.2	Application of Eqn. (2.42) to different repair distributions	78
3.1	Minimal cut set generated by BIT for example fault tree	97
3.2	Comparison of results of this study with the results of KITT for an example fault tree	120
3.3	Summary of important characteristics of a selected distribution	131
4.1	AFWS component failure characteristics data	145
4.2	Component's importance calculations of AFWS using Birnbaum and Fussell-Vesely measures	149
4.3	Aux-Feed Water Component's average unavailability error factor (EF)	150
4.4	AFWS cumulative unavailability distribution	151
4.5	AFWS component failure characteristics error factors data	156
4.6	AFWS component failure characteristics error factors data, an example for long tail log-normal and gamma distributions	159
4.7	The effect of the Monte-Carlo trials on the top event distribution	160
4.8	The effect of the Monte-Carlo trials on the top event distribution for mixed components failures distributions	161

LIST OF TABLES (Cont.)

<u>Table</u>	<u>Title</u>	<u>Page</u>
4.9	Importance analysis of AFWS components given that one of the pump train is out of service	162
4.10	Component's parameters data for the electrical example	168
4.11	CVCS pump trains component's parameters data	175
4.12	Comparison of the CVCS unavailability calculations for the exponential and second order erlangian repair distribution	177
A.1	Comparison of cumulative probability distribution of a log-normal with $\mu = 3.0445$ and $\sigma = 0.57$ with its approximated combination of exponentials with $\lambda_1 = 6.319 \times 10^{-2}$ and $\lambda_2 = 1.1263 \times 10^{-1}$	217
A.2	Comparison of the cumulative probability distribution of a log-normal with $\mu = 3.0445$ and $\sigma = 0.2$, with its approximated combination of exponentials, with $\lambda_1 = 8.1783 \times 10^{-2}$ and $\lambda_2 = 1.0873 \times 10^{-1}$	218
A.3	Comparison of the cumulative probability distribution of a log-normal, with $\mu = 4.60517$ and $\sigma = 0.66785$, with its approximated combination of exponentials, with $\lambda = 1.0788 \times 10^{-2}$ and $\lambda_2 = 3.0955 \times 10^{-2}$	219
B.1	List of Boolean operations and identities	231
B.2	List of minimal cut sets generated by BIT for Fig. B.2	234
B.3	List of minimal cut sets generated by BIT for Fig. B.3	237

LIST OF TABLES (Cont.)

<u>Table</u>	<u>Title</u>	<u>Page</u>
B.4	Equivalent failure and repair rate of an AND/OR gate, for monitored components and components with constant unavailability	239
C.1	List of input to UNRAC for the sample problem	270
C.2	Results of UNRAC for the sample problem	272

CHAPTER 1: INTRODUCTION1.1 Reliability Concept and Methods of Evaluations

Reliability analysis is a method by which the degree of successful performance of a system under certain stipulated conditions may be expressed in quantitative terms. In order to establish a degree of successful performance, it is necessary to define both the performance requirement of the system and the expected performance achievement of the system. The correlation between these two can then be used to formulate a suitable expression of reliability as illustrated in Fig. 1.1, Green and Bourne (1972).

Since both the required and achieved performance may be subject to systematic and random variations in space and time, it follows that the appropriate reliability expression is generally of a probabilistic nature. Most definitions of reliability, of which the following is the standard form approved by the IEEE, takes this aspect into account; (IEEE.STD352 (1972))

"Reliability is the characteristic of an item expressed by the probability that it will perform a required function under stated condition for stated period of time."

The word "item" can cover a wide range in its use, i.e., it could mean a single pressure switch or a large system.

The first step in a reliability analysis, therefore, is to ascertain the pattern of variation for all the relevant

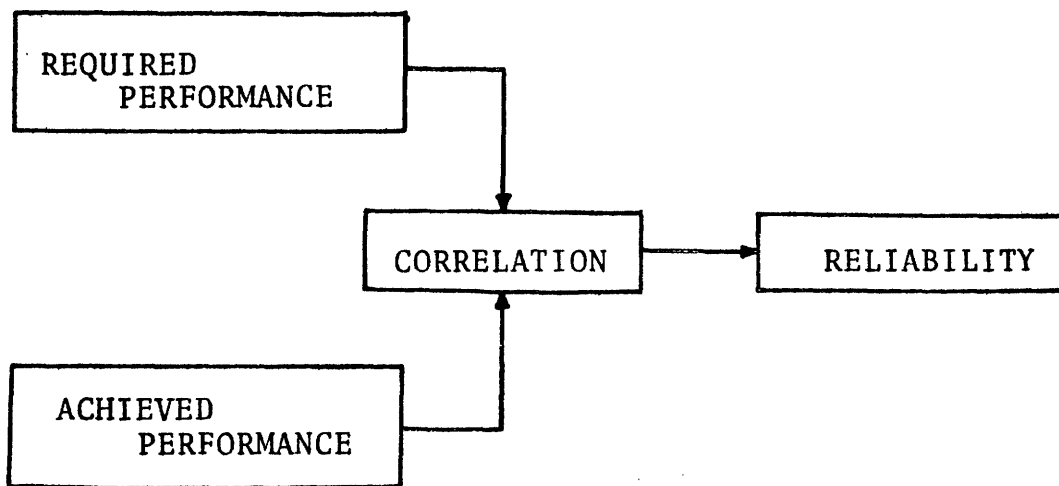


Fig. 1.1: Overall reliability concept.

performance parameters of the system both from the point of view of requirement as well as from likely achievement. Performance variations may be due not only to the physical attributes of the system and its environment but also the basic concepts, ideas and theories which lie behind the system's design. Having established all the appropriate patterns of variations, the system should be rigorously examined to check its ability to work in the required way or fulfill the correct and safe overall function. To do so, the system should be analyzed as to when and how it may fail. Failures may be partial or catastrophic which results in partial or complete loss of operation or performance. Failures may affect individual components or may cascade from one component to another. Each type of failure will generally lead to a different chain of events and a different overall result. In order to establish the appropriate probability expressions for each chain of events, it is useful to convert the system functional diagram into a logic sequence diagram. There are two ways to develop the system logic diagram.

- 1) positive logic
- 2) negative logic

In the positive logic, the chain of events consists of a series of operational components or sub-systems that constitute successful operations whereas in the negative logic the chain of events are composed of component or sub-system failure. The positive logic is called reliability block

diagram which has been used extensively during early designs of satellites. The negative logic has acquired different names depending on the types of logic interconnections.

They are:

- 1) fault tree logic
- 2) event tree logic
- 3) cause consequence chart logic

Reliability analysis by using the method of fault tree is the most well known analytic method in use today for studying the failure modes of complex systems. Of course, event tree and cause consequence chart have also been used, however, by proper rearrangement, one can argue that the cause-consequence chart (ccc) can be mapped into event-tree and fault tree combinations and both can be mapped into a set of fault trees for which the Top events are the consequences of the ccc or event tree, Evans (1974), Taylor (1974).

1.2 Fault Tree and Reliability Analysis

According to Haasl (1965), the concept of Fault Tree Analysis (FTA) was organized by Bell Telephone Laboratories as a technique with which to perform a safety evaluation of the Minuteman Launch Control System in 1961. Bell engineers discovered that the method used to describe the flow of "correct" logic in data processing equipment could also be used for analyzing the "false" logic which results from component failure. Further, such a format was ideally suited to the

application of probability theory in order to numerically define the critical fault modes.

From the time of its invention, FTA has been used successfully to assess, qualitatively and/or quantitatively, different safety studies involving complex systems. The analysis begins with a well defined undesired event (failure of the system to perform its intended function under given condition) which is called the Top event. Undesired event is identified either by inductive analysis or a failure mode and effect analysis. In that, one proceeds to uncover the various combination of 'simpler' events (i.e., sub-system and component failure) whose occurrence guarantees the occurrence of the Top event. The analysis stops at the level at which the events are completely understood and quantitative information regarding their frequency of occurrence is available. These events are called the primary input of the tree. Figure 1.2 shows a typical structuring of a fault tree.

The reliability analysis in safety studies of nuclear reactors using fault tree analysis was advocated in the early 1970's, Garrick (1970). In 1975, a unique and comprehensive safety study on nuclear power plants using FTA was completed, WASH 1400 (1975). In that study, the potential risk of nuclear power plants to the public in the U.S. was estimated. Many other countries have since used the same methodology as in WASH 1400 to estimate their power plant risk, Ringot (1978). Nuclear Engineering (1979). Today, fault tree is widely used

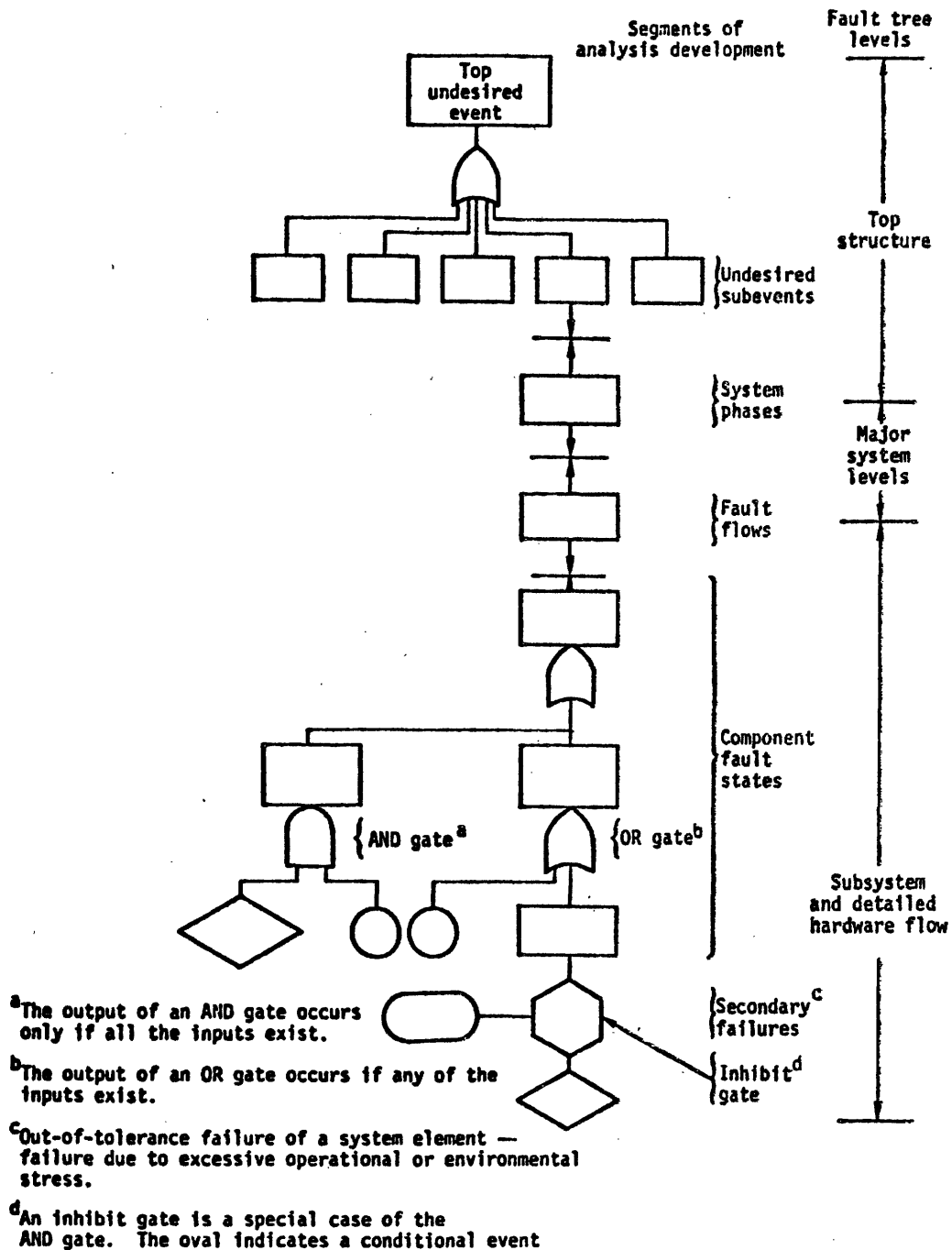


Fig. 1.2: Typical structuring of a fault tree (Barlow and Lambert (1975)).

as a principle structure for performing system reliability calculations, i.e., risk analysis of advanced reactors (Liquid Metal Fast Breeder, LMFBR), Burdick et al. (1977), Waste Management Accidental Risk, Battelle (1976) etc. In summary, fault tree analysis has been used extensively in reliability analysis since its invention in 1961. Therefore, in this study, we use fault tree to estimate the reliability of the given system. Before entering the evaluation techniques of fault tree, we refer the reader to the "Glossary of Words" given at the end of this Chapter.

1.3 Fault Tree Evaluation

Fault tree evaluation refers to the analysis of the system logic model provided by the fault tree. This analysis can be either qualitative or quantitative or both.

1.3.1 Qualitative Analysis

Qualitative analysis is comprised of those analytical procedures that do not require the assignment of probability values to any of the events defined on the system logic. Qualitative analysis tends to deal with the problem on a more basic level and to identify fundamental relationships that can be established from logic model without quantification. An important aspect of qualitative analysis is to determine the minimal cut sets for the Top event or some intermediate event. The importance stems from the fact that determining minimal cut sets is not only an analytic goal itself, i.e., finding all of the fundamental ways that an event can occur, but it is a

required initial step in many other fault tree evaluation techniques including quantification. Several computer programs have been designed to achieve this important goal. In general, the codes have been developed in one of two ways. In one development, the concept of Boolean Algebra has been adapted -- Bennett (1975), Fussell et al. (1974b) -- i.e., SETS by Worrel (1975) and ELRAFT by Semanderes (1971), and in the other the concepts of cut and path sets have evolved from Coherent Structure Theory, Barlow, Proschan (1975), Vesely (1970), i.e., PREP by Vesely and Narum (1970) and MOCUS by Fussell, et al. (1974a)

Thus, a prime implicant of the Boolean function defined by an equation of system behavior, Bennett (1975) and Henely and Kumamoto (1978), corresponds to a minimal cut set obtained from a system structure function. Differences as well as similarities occur in these approaches, but both are appropriate as long as an event and its complement cannot both occur in each set.

Other codes in this category are: MODCUT (Modarres (1979)), WAM-CUT (Erdmann (1978)), ALLCUTS (Van Sylke and Griffing (1975)) and TREEL and MICSUP (Pande et al. (1975)).

1.3.2 Quantitative Analysis

As stated earlier, reliability analysis is of probabilistic nature therefore, a complete quantification of the system is required in order to be able to assess a meaningful value for the reliability of the system. In the fault tree analysis,

since the system structure logic is composed of a series of negative (failure) logic, the term reliability is always replaced by the term "unreliability." In a quantitative sense, unreliability is a complement value of reliability. As discussed in the previous section, generating the minimal cut sets is the first step in any FTA. The second step in FTA is to find the Top event unreliability by proper assignment of probability values (data) to each basic events (components failure).

The assignment of data described above will depend on the type of results required. For example, if a point estimate of the Top event failure probability is to be determined, then the point estimates for the component failure probabilities (or data allowing their computation) will have to be assigned. Similarly, if a distribution is to be found for the Top event unreliability, then one or more of the component characteristics will have to be assigned in terms of a distribution.

Given the above data, the following quantitative evaluations are generally useful in assessing a system reliability:

Numerical Probabilities: Probabilities of system and/or component failure (point estimate and/or time dependent

Quantitative Importance: Quantitative ranking of components or cut sets to system failure

Sensitivity Evaluation: Effect of change in models and data, error bounding (multiple

point estimates, confidence interval, distribution).

Having performed the above evaluations, a designer and/or a user can compare the given systems and determine which will probably be more dependent and have higher unreliability. Hence, quantitative analysis is of great importance in choosing the better system.

Several codes have been written for such evaluations. Among them in FORTRAN are: KITT 1 and 2 (Vesely and Narum (1970)), FRANTIC (Vesely and Goldberg (1977)), SAMPLE (WASH 1400 (1975)), LIMITS (Lee and Salem (1978)) and in PL/1 are: PL-MODT and PL-MODMC by (Modarres (1979)). The above mentioned FORTRAN codes need minimal cut sets or related system structure function as input, and each of them has certain limitations in their applicability to different systems. For example, although KITT is always coupled with PREP for its minimal cut sets requirement, the coupled code is unable to accommodate the periodically tested components as input and to evaluate error bounds on the Top event. On the other hand, the FRANTIC code can accommodate all types of monitoring situations, i.e., non-repairable, repairable, periodically tested etc., but it requires the system structure function as input and it is incapable of evaluating the error bound. In addition, the approximations used in the code greatly overestimates the system unreliability when the failure rates are greater than 10^{-3} per hour. Finally, SAMPLE and LIMITS can only be used for

error propagation given the average unreliability of each event and require the system structure function as an input. In the case of PL-MOD series codes, both codes should be used to calculate all three aforementioned evaluations.

1.4 Objective of Thesis

To assess the reliability of different designs, one has to perform a complete qualitative and quantitative analysis of the respective designs. The probabilistic values for the comparison are the evaluations that were discussed in the previous section. Several codes have been written by different authors to do such calculations, (see Section 1.3.1 and 1.3.2). Unfortunately, to do a complete job one has to use two or three different codes depending on the type of results desired. Therefore, the chance of making an error in the preparation of the input data will greatly increase. Hence, there is an incentive to prepare a single package composed of different codes in order to minimize the confusion and error that arise in using combination of the codes.

The objective of this thesis is to develop a code which can evaluate the system unreliability (point estimate and/or time dependent), find the quantitative importance of each component involved, and calculate the error bound on the system unreliability evaluation. To prepare such a code, the methodologies used in REBIT (Wolf (1975)), KITT (Vesely (1970)), FRANTIC (Vesely and Goldberg (1977)), SAMPLE (WASH 1400 (1975)), LIMITS (Lee and Salem (1978)) codes were implemented.

For generating minimal cut sets, a portion of REBIT which is called BIT is used. BIT, at the beginning, was limited to 32 components and gates and only would accept AND or OR gates logic. For this study, it has been modified to accept any kind of gate logic and its capability has been increased at the present time to 250 components and/or gates. However, this capability could be increased to any number required. For the quantification, a general and consistent set of mathematical models for the component failure were developed and FRANTIC code was restructured to accommodate these models. For the components' important analysis, the differentiating method (the methods of Birnbaum) and Fussell-Vesely methods (Lambert (1975)) were implemented. Finally, for error propagation analysis, we developed a routine that allows the simulation to be carried out on the component's failure parameters (failure rate, repair rate, test rate, etc.) as well as on the average components unreliability.

1.5 Structure of Thesis

Chapter 2 discusses the basic concept of the reliability analysis. In particular, it will show how different mathematical models can be used in the calculation of the time dependent reliability and/or unreliability of a system or component.

Chapter 3 explains the principle structure of the code developed in this study to accommodate all the models discussed in Chapter 2. The code calculates the unreliability of any system given its fault tree and is called Unreliability

Analysis Code (UNRAC). In addition, it will discuss how the method of moment matching will be helpful in approximating the Top event distribution. Furthermore, two methods of accommodating the dependencies in the fault tree will be discussed.

Chapter 4 presents a series of examples to insure the effectiveness of models encoded and compares some of the WASH 1400 results with the time dependent values calculated by this study.

Finally, Chapter 5 contains the summary and conclusion of this study and it will discuss the appropriate recommendation for further studies.

1.6 Glossary of Words

In this section, we present a list of standard terminology that is in use today for any fault tree analysis. Most of the symbols and terminology defined herein are taken from IEEE-STD-352 (1972).

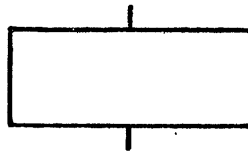
1. Basic Event of Fault: Basic Fault event is an event that requires no further development. In the fault tree, Basic event is represented by a circle. The probability of such event is derived from empirical data or physics of failure analysis.



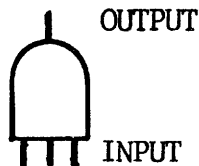
2. Faults Assumed Basic to the Given Fault Tree: This type of event is shown by a diamond in a given fault tree. The possible causes of this event are not developed either because it is not required or the necessary information is unavailable.



3. Resultant Events: The rectangle represents an event that results from the combination of events of the type described above through the input of a logic gate.

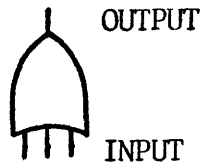


4. AND Gate: AND gate describes the logical operation whereby the coexistence of all input events is required to produce the output event. In other words, an output event occurs if and only if all the input occurs.

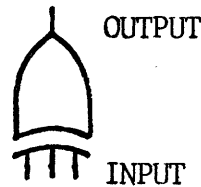


5. OR Gate: OR gate defines the situation whereby the output event will exist if one or more of the input event exists. That is, an output event occurs if one

or more of the input occur.



6. Exclusive OR (EOR) Gate: This gate functions as an OR gate with the restriction that specified inputs cannot coexist.



7. Transfer Gates: The transfer symbol provides a tool to avoid repeating sections of the fault tree. The "Transfer out" gate represents the full branch that follows it. It is represented by a symbol, say b, and indicates that the branch is repeated somewhere else. The "Transfer in" gate represents the branch (in this case b) that is already drawn somewhere else, and instead of drawing it again, it is simply input at that point. The transfer gate is shown by a triangle in the fault tree.



8. Failure: The termination of the ability of an item to perform its required function. In general, there are two types of failure:

- a) Revealed (announced) failure. A failure of an item which is automatically brought to light on its own occurrence.
 - b) Unrevealed failure. A failure of an item which remains hidden until revealed by some thorough proof-testing procedure.
9. Failure Rate: The expected number of failures of a given type, per item, in a given time interval.
10. Mutually Exclusive Events: Events that cannot exist simultaneously.
11. Repair Rate: The expected number of repair actions of a given type completed per unit of time.
12. Test Interval: The elapsed time between the initiation of identical tests on a same item.
13. Test Schedule: The pattern or testing applied to the parts of a system. In general, there are two patterns of interest:
- a) Simultaneous: Redundant items are tested at the beginning of each test interval, one immediately following the other.
 - b) Perfect Staggered: Redundant items are tested such that the test interval is divided into equal sub-intervals.

14. Logic Diagram: A diagram shown in a symbolic way the flow and/or processing of information.
15. Primary Failure: Basic event failure is called Primary failure.
16. Cut Set: is the set of the basic event whose simultaneous failure guarantees the top event failure.
17. Path Sets: is the set of the basic event whose success guarantees the system success.
18. Minimal Cut Set (Mode failure, Critical Path): is the smallest set of primary failures such that if all these primary failures simultaneously exist, then the Top event failure exists.
19. Secondary Failure: is a failure due to excessive environmental or operational stress placed on the system element.
20. Confidence Level: the probability that the assertion made about the value of the distribution parameter, from the estimates of the parameter, is true.

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

CHAPTER 2: ON THE ESSENTIAL OF RELIABILITY & PROBABILITY THEORY

2.1 Introduction and Basic Concepts

System reliability is a measure of how well a system performs or meets its design objective. If successful operation is desired for a specified period of time, reliability is defined as the probability that the system will perform satisfactorily for the required time interval.

Historically, reliability has usually been considered during system design, but it acquired more attention first in radar design in early 1940 and then in design of the Minuteman and satellite systems. Most of the mathematical models used to help better understand system reliability have been studied by the pioneers in this field and their applications have been tested during the aforementioned designs. This work is an effort to further develop the analytical methods that have been established earlier by such authors as: Cramer (1946), Feller (1957), Parzen (1960), Cox (1963) and Takacs (1957).

Therefore, in this Chapter, we discuss the essentials of reliability and probability theory that will be used in developing the model in succeeding Chapters. To do this, we first define the measure(s) of system effectiveness and then discuss the appropriate theory and mathematical model to determine these measures.

In general, the measures of system effectiveness are criteria by which alternate design policies can be compared and judged, i.e., alternate design policies for an Emergency Safety feature may be judged on the probability of actuation detection and/or instantaneous availability of the system. However, there are many problems associated with developing measures of effectiveness. In the first place, it is not always possible to express quantitatively all the factors of interest in a model. Secondly, the measures of the system's effectiveness can change with time. What may be effective today may not be tomorrow due to technological and political innovations. Today there are four major measures of system reliability effectiveness. These are:

1) Availability

This measure is applicable to maintained systems.

There are several measures of availability in use today, which we can categorize as:

- a) Instantaneous Availability: The probability that the systems will be available at any random time t .
- b) Average Up-Time: The proportion of time in a specified interval $(0, T)$ that the system is available for use.
- c) Steady-State Availabilitiy: The proportion of time the system is available for use when the time interval considered is very large.

In the limit, a) and b) approach the steady-state availability c). Which measure to choose depends upon the system mission and its condition of use.

2) Probability of Survival

The probability of system survival is a measure of the probability that a system will not reach a completely failed state during a given time interval, given that at the beginning of the interval the system was in a fully operable state. The appropriate use for the probability of survival as a measure of system reliability effectiveness occurs in those systems where maintenance is either not possible during operation or can only be performed at defined times, i.e., equipment in the reactor building. This measure is highly dependent upon system configuration and the condition of maintenance. In general we would expect a doubly redundant system to have a higher survival probability than a non-redundant system. Also, we should expect to have a higher survival probability where repair on a redundant system is being performed immediately upon the inception of the failure of one of the parallel equipment trains, than where repair is not begun until all equipment reaches a failed state. We shall henceforth denote the latter case as a non-maintained and the former case as a maintained system.

3) Mean Time to System Failure

The literature on reliability contains such terms as Mean-Time-Between-Failure (MTBF), Mean-Time-To-Failure (MTTF) and Mean-Time-To-First-Failure (MTTFF). These terms do not describe the same thing. MTBF is specifically applicable to a population of equipment where we are concerned with the average time between the individual equipment failures. When we are interested in one component or system, the MTTF and MTTFF are applicable. The difference between MTTFF and MTTF is the specification of the initial condition and how time is counted. The difference becomes apparent when dealing with redundant systems under different maintenance conditions. For non-maintained system, MTTF is a measure of expected time the system is in an operable state before all the equipment reaches a failed state and the time is counted from initial fully operable to final failed state. For a maintained system, MTTFF is a measure of the expected time the system is in an operable state given that all equipment was initially operable at time zero.

4) Duration of Single Downtime

For some systems, the duration of a single downtime may be the most meaningful measure of system reliability effectiveness. For example, an air traffic control system which establishes flight plans and

directs landing and take-off, a long duration of single downtime may mean queuing at take-off, schedule delays, lost flights and so on. In reactor safety, long downtime of an emergency system may require the shut down of the reactor.

In selecting a measure of effectiveness, we should keep in mind that the measure will have little value if one is unable to express it quantitatively in order to compare between alternate designs. To be able to express these measures quantitatively, we need to model each equipment or component of a system according to its maintenance policy. Each component can come under the following two major categories:

- 1) non-maintained
- 2) maintained.

In one approach, Vesely and Goldberg (1977), the components are classified according to the type of failure rates and the manner in which repair policies are executed. The Vesely and Goldberg (1977) classification is as follows:

- Class 1: Component with constant unavailability
- Class 2: Component with unrepairable policy (non-maintained)
- Class 3: Component with repair in which the faults are immediately detected (maintained)
- Class 4: Component with repair in which the faults are detected upon inspection, i.e., periodically

maintained component (maintained).

As we can see, there is not much difference between the above classification and the two major repair policies. Therefore, in the following sections we discuss the mathematical models for non-maintained and maintained components.

2.2 Reliability Model of Non-Maintained Systems

When a fixed number, N , of identical systems are repeatedly operated for t amount of time, there will be $N_s(t)$ systems that survive the test and $N_F(t)$ that fail. The reliability, $R(t)$, of such a collection of systems can be expressed as:

$$R(t) = \frac{N_s(t)}{N_s(t) + N_F(t)} \quad (2.1)$$

or equivalently

$$R(t) = 1 - \frac{N_F(t)}{N} \quad (2.2)$$

letting N be fixed we can write

$$\frac{dN_F(t)}{dt} = - N \frac{dR(t)}{dt} \quad (2.3)$$

where $\frac{dN_F(t)}{dt}$ is the rate at which systems fail. Now if we divide both sides of Eqn. (2.3) by $N_s(t)$, and denote the left hand side by $r(t)$

$$r(t) = \frac{1}{N_s(t)} \frac{dN_F(t)}{dt} = - \frac{N}{N_s} \frac{dR(t)}{dt} \quad (2.4)$$

then $r(t)$ is the instantaneous probability of failure. By simple manipulation of Eqn. (2.4) we get

$$R(t) = \exp \left(- \int_0^t r(x) dx \right) \quad (2.5)$$

The function $r(t)$ is called the failure rate function, hazard function or hazard rate.

A study of many systems (Mechanical, Electronic, (Davis (1952))), during their normal life expectancy has led to the conclusion that their failure rate follows a certain basic pattern. (See Fig. 2.1)

It has been found that systems exhibit a high failure rate during their initial period of operation. These are failures which are typically caused by manufacturing flaws and this initial period is called the infant mortality or break in period. The operation period that follows infant mortality has a smaller failure rate which remains approximately constant until the beginning of the Wear-out period. Most of the components in use are considered to have constant failure rates and the failures generally result from severe, unpredictable and usually unavoidable stresses that arise from environmental factors such as vibration, temperature shock, etc. Therefore, in this study we assume a constant failure rate $r(t) = \lambda$, for any component. Hence from Eqn. (2.5) we have

$$R(t) = \exp (-\lambda t) \quad (2.6)$$

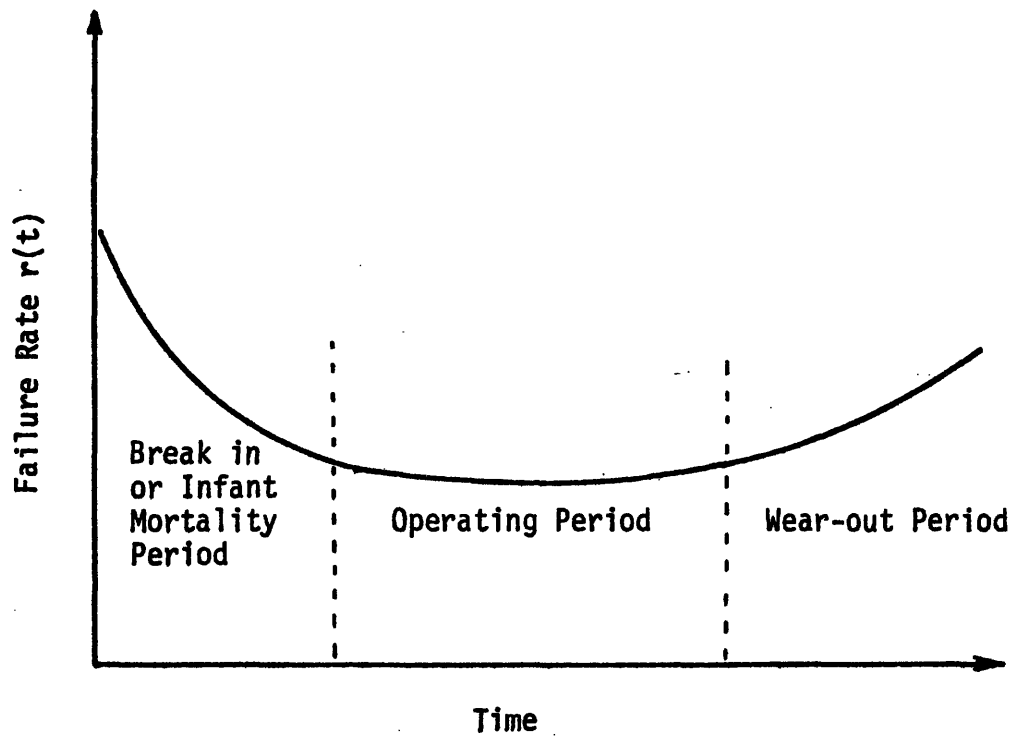


Fig. 2.1: Typical failure rate behavior.

and if we assume that the system or the component under study has a failure density function of $f(t)$ and failure distribution function $F(t)$ we have

$$f(t) = \frac{dF(t)}{dt} = - \frac{dR(t)}{dt} = \lambda \exp(-\lambda t)$$

where $f(t)$ is an exponential failure time density function. Therefore, for a non-maintained component or system, the reliability function is given by Eqn. (2.6) and unreliability by:

$$F(t) = 1 - R(t) = 1 - \exp(-\lambda t) \quad (2.7)$$

and the mean time to failure (MTTF) is given by:

$$MTTF = \int_0^{\infty} t f(t) dt = 1/\lambda$$

2.3 Reliability Model of Maintained Systems

2.3.1 Introduction

When we allow repair during the desired interval of operation, the concept of reliability is usually replaced by the concept of instantaneous availability which is defined as the probability that the system is in an acceptable state at any time, t , given that the system was fully operable at time $t = 0$. Another important measure in a maintained system is system downtime, which is defined as the total time from

system failure until return to service. Service or repair downtime is generally divided into two basic categories:

- 1) Active repair time
- 2) Administration time

Active repair time can be divided into recognition or detection time, fault location or diagnosis time, correction or repair time and verification or final malfunction check time. Administrative contribution to downtime is that time required to obtain a spare part, or is spent in waiting for maintenance crew, tools or test equipment. The time required to perform each of these categories varies statistically from one failure instance to another and there usually exists a large number of relatively short-time and a small number of long-time repair periods. However, in modeling a maintained system, the repair downtime is frequently represented by one of the following probability density functions:

- 1) exponential
- 2) log normal
- 3) gamma

In this study we assume $g(t)$ to represent the general repair time density function, the mean downtime or Mean-Time-To-Repair (MTTR) is denoted by:

$$MTTR = \int_0^{\infty} t g(t) dt \quad (2.8)$$

and the probability that a failed system returns to service by some time T is given by:

$$P_r \{ \text{downtime} \leq T \} = \int_0^T g(t) dt \quad (2.9)$$

For evaluating the system unavailability one can use the following two methods:

- 1) differentiating method applicable for simple failure and repair distribution
- 2) Semi-Markovian method for more complex distribution and system.

In this section, we discuss the differentiating method for a two-state system with failure and repair time density functions. At any time, t , the system is in one of the following states with probability, say, $P_i(t)$, $i = 1, 2$.

State 1: The system is up

State 2: The system is down

$g(t) = \mu e^{-\mu t}$ repair distribution where μ is the repair rate.

If the system is in state 1 at time t , the probability that the system does not go to state 2 before $t + \Delta t$ is equal to the reliability of the system and is:

$$e^{-\lambda \Delta t} = 1 - \lambda \Delta t + \underline{O}(\Delta t),$$

where $\underline{O}(\Delta t)$ is a quantity having the property that

$\lim_{\Delta t \rightarrow 0} \frac{\underline{O}(\Delta t)}{\Delta t} = 0$. If the system is in state 2 at time t ,

the probability that the system remains in state 2 before $t + \Delta t$ is equal to (from Eqn. (2.9)) $e^{-\mu\Delta t} = 1 - \mu\Delta t + \underline{O}(\Delta t)$.

Therefore, in order for the system to be in state 1 at some time $t + \Delta t$, either it must be in state 1 at time t and no failure occurs in the interval $(t, t + \Delta t)$, or it is in state 2 at time t and it is repaired during $(t, t + \Delta t)$. Thus:

$$P_1(t + \Delta t) = P_1(t)[1 - \lambda\Delta t] + P_2(t)\mu\Delta t + \underline{O}(\Delta t)$$

$$\text{or } \frac{P_1(t + \Delta t) - P_1(t)}{\Delta t} = \mu P_2(t) - \lambda P_1(t) + \frac{\underline{O}(\Delta t)}{\Delta t}$$

letting $\Delta t \rightarrow 0$

$$P_1'(t) = \mu P_2(t) - \lambda P_1(t) \quad (2.10)$$

where

$$P_1'(t) = \frac{dP_1(t)}{dt}$$

But we know that $P_1(t) + P_2(t) = 1$

$$\therefore P_1'(t) = \mu - (\lambda + \mu)P_1(t) \quad (2.11)$$

Eqn. (2.11) is a simple differential equation with initial conditions of $P_1(0) = 1$, $P_2(0) = 0$, Hence

$$P_1(t) = \frac{\lambda}{\lambda+\mu} + \frac{\mu}{\mu+\lambda} \exp[-(\lambda+\mu)t]$$

The instantaneous availability of the system is equal to $P_1(t)$:

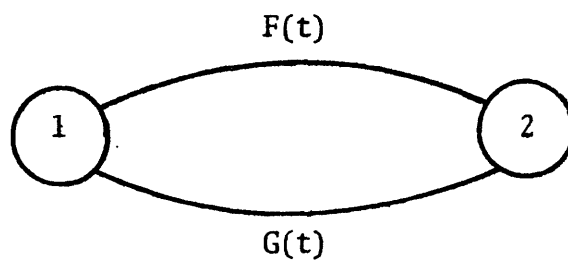
$$A(t) = P_1(t) = \frac{\mu}{\lambda+\mu} + \frac{\lambda}{\mu+\lambda} \exp[-(\lambda+\mu)t] \quad (2.12)$$

The term steady state availability is used when the mission period is much larger than MTTR.

$$A_{ss}(t) = A(\infty) = \frac{\mu}{\mu+\lambda} = \frac{MTTF}{MTTR + MTTF} \quad (2.13)$$

Note: In the above discussion we assume the unit (system) is as good as new after repair or replacement.

In a more general case, the method of repair as a renewal process has been used by Barlow and Hunter (1961), Cox (1963), Takacs (1959) to evaluate the up and down state probabilities of a unit (system) by developing a detailed mathematical model to inter-relate failure and repair distributions to the probabilities. To prevent duplication of the method, we only show the modified result of state and transition probabilities for a two-state system. Let $F(t)$ be the failure probability distribution and $G(t)$ be the repair probability distribution. By virtue of the state diagram shown in Fig. 2.2 one can write: (for proof see Rau (1970)).



state 1: is Up State

state 2: is Down State

$F(t)$: Failure Probability Distribution

$G(t)$: Repair Probability Distribution

Fig. 2.2: A simple two-state transition diagram.

$$P_{12}^*(s) = \frac{F^*(s) [1 - G^*(s)]}{S[1 - G^*(s)F^*(s)]} \quad (2.14)$$

$$P_{21}^*(s) = \frac{G^*(s) [1 - F^*(s)]}{S[1 - G^*(s)F^*(s)]} \quad (2.15)$$

$$P_{11}^*(s) = \frac{1 - F^*(s)}{S[1 - G^*(s)F^*(s)]} \quad (2.16)$$

$$P_{22}^*(s) = \frac{1 - G^*(s)}{S[1 - G^*(s)F^*(s)]} \quad (2.17)$$

where $K^*(s)$ is the Laplace transform of $K(t)$, $K=f,g$

$P_{ij}^*(s)$ is the Laplace transform of $P_{ij}(t)$, $i,j=1,2$

$P_{ij}(t) = P_r$ {system is in state j at time t given in state i at time 0}

If we assume exponential distribution for both failure and repair, the inverse transform of equation (2.16) will yield Eqn. (2.12).

If $f(t) = \lambda e^{-\lambda t}$ and $g(t) = \beta$ (constant repair) then one can show

$$P_{11}(t) = \sum_{j=0}^{[t/\beta]} \frac{\lambda^j (t - \beta j)^j e^{-\lambda(t - \beta j)}}{j!} \quad [2.18]$$

where $[t/\beta]$ is the largest integer less than or equal to (t/β) .

One should note that it is not always easy to find the inverse transform of $P_{ij}^*(s)$ except for special cases (which we mentioned above) and typically one must use numerical

technique to find $P_{ij}(t)$.

2.3.2 Semi-Markovian Model

In the last section, we discussed a method which is only applicable to a two-state system. In general, a system is comprised of different components and each component is subject to a certain inspection-repair-replacement policy. Therefore, a more general model is needed to handle a complex system. One of the models which has been used successfully in the past two decades is the Markovian model. The application of this model in reliability analysis can be found in numerous references, some recent ones being: Kemeny and Snell (1960), Barlow and Proschan (1965), Shooman (1968), Green and Bourne (1972). In this section, we utilize the model discussed by Howard (1972). Basically, Markovian models are functions of two random variables; the state of the system i and the time of observation t . A discrete parameter stochastic process $\{X(t); t=0,1,2,\dots\}$ or a continuous parameter process $\{x(t); t \geq 0\}$ is said to be a Markov process if, for any set of n time points, t , where $t_1 < t_2 < \dots < t_n$, in the index set of the process and any real numbers, $x_1, x_2, \dots, x_n$;

$$\begin{aligned} P_r[X(t_n) \leq x_n | X(t_1) = x_1, \dots, X(t_{n-1}) = x_{n-1}] \\ = P_r[X(t_n) \leq x_n | X(t_{n-1}) = x_{n-1}] \end{aligned} \quad (2.19)$$

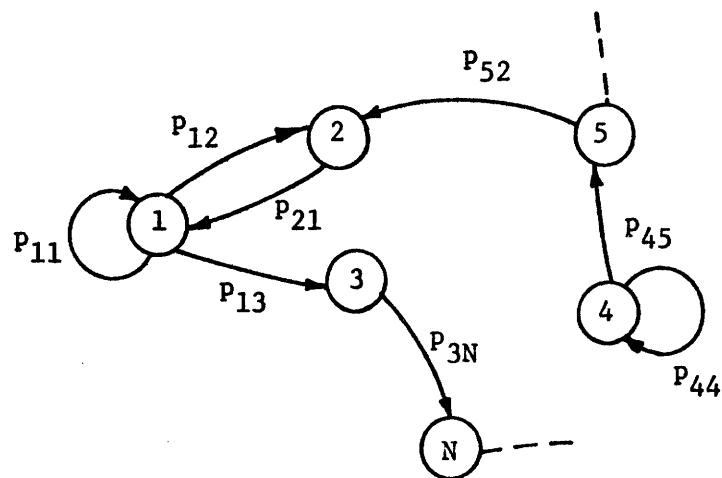


Fig. 2.3: A typical transition diagram.

Central to the theory of Markov process models are the concepts of state and state transition. In reliability analysis, a state can often be defined simply by listing the components of a system which are working satisfactorily. And, in general, the number of distinguished states depends on the number and function of the system equipment. In the model we will always consider a finite number of states.

The manner in which the states are interchanged in the course of time is called transition. As we can see from the general Markov process, equation (2.19), only the last state occupied by the process is relevant in determining its future behavior. Thus, the probability of making a transition to each state of the process depends only on the state presently occupied. If we denote p_{ij} to be the transition probability, by definition we have:

$$p_{ij} = P_r\{S(n+1) = j | S(n) = i\} \quad 1 \leq i, j \leq n \quad (2.20) \\ n=0, 1, 2, \dots$$

and it is defined as the probability that a process presently in state ($S(\cdot)$) i will occupy state j after its next transition, see Fig. 2.3.

Another quantity which is always needed in dealing with a Markov process is the state transition probability denoted by $\phi_{ij}(n)$ and is defined to be the probability that the process will occupy state j at time n given that it occupied state i

at time 0 , or

$$\phi_{ij}(n) = P_r \{S(n) = j | S(0) = i\} \quad 1 \leq i, j \leq N \quad (2.21)$$

$$n=0, 1, 2, \dots$$

Both p_{ij} and $\phi_{ij}(n)$ have the following properties:

$$\begin{aligned} 1) \quad & 0 \leq \phi_{ij}(n), p_{ij} \leq 1 \quad 1 \leq i, j \leq N \\ 2) \quad & \sum_{j=1}^N \phi_{ij}(n) = 1 \quad i, j=1, 2, \dots, N \quad (2.22) \\ & n=0, 1, 2, \dots \\ & \sum_{j=1}^N p_{ij} = 1 \end{aligned}$$

In the above equations N is the total number of states and n is the number of transition steps in time.

Having defined Markov processes, we can relate the aforementioned definitions to the continuous time Semi-Markov process. The Semi-Markov process is a process whose successive state occupancies are governed by the transition probability of a Markov process, but whose stay in any state is described by a positive random variable, τ_{ij} , that depends on the state presently occupied, and the state to which the next transition will be made, j . The quantity τ_{ij} is called holding time and has a holding time density function $h_{ij}(t)$. One can also define an unconditional waiting time density function of state $w_i(t)$ by

$$w_i(t) = \sum_{j=1}^N p_{ij} h_{ij}(t) \quad (2.23)$$

for the cases when we do not know the successor state. By virtue of equations (2.20), (2.21), and (2.23) one can write:

$$\phi_{ij}(t) = \delta_{ij} {}^{cc}w_j(t) + \sum_{k=1}^N p_{ik} \int_0^t d\tau h_{ik}(\tau) \phi_{kj}(t-\tau) \quad (2.24)$$

$i, j=1, 2, \dots, N \quad t > i$

Where

$\phi_{ij}(t)$ = is the probability that the process will occupy state j at time t given that it entered state i at time 0 .

δ_{ij} = Kronecker delta = $\begin{cases} 1 & i=j \\ 0 & i \neq j \end{cases}$

${}^{cc}w_i(t)$ = cumulative complementary distribution of $w_i(.)$ and is the probability that the process will leave its starting state i at time greater than t .

The physical meaning of terms in Eqn. (2.24) can be defined as:

first term = Probability that the process will end up in state j by one step during $(0, t)$

second term = Probability that the process makes its first transition from the starting state i to some state k at some time $\tau, 0 \leq \tau \leq t$, and then proceeds somehow from state k to state j in the remaining time $t - \tau$. This probability is summed over all states k and time 0 to t .

One can solve the Equation (2.24) by two methods:

- 1) Method of matrix manipulation
- 2) Method of flow graph theory

In this section we discuss the first method. If we take the Laplace transform of equation (2.24) we will have:

$$\phi_{ij}(s) = \delta_{ij} {}^{cc}w_i(s) + \sum_{k=1}^N p_{ik} h_{ik}(s) \cdot \phi_{kj}(s) \quad (2.25a)$$

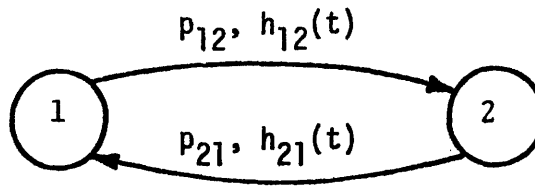
and in matrix form

$$\Phi(s) = [I - PH(s)]^{-1} {}^{cc}W(s) \quad (2.25b)$$

Where:

$$\begin{aligned} \Phi(s) &= (\phi_{ij}(s)) & I &= \text{identity Matrix} \\ P &= (p_{ij}) \\ H(s) &= (h_{ij}(s)) \\ {}^{cc}W(s) &= ({}^{cc}w_i(s)) \end{aligned}$$

The right hand side of equation (2.25) consists of known functions of s and one can find $\Phi(s)$ analytically or numerically depending on the degree of complexity of the holding time distributions. To demonstrate the application of equation (2.25) we consider a general two-state unit with holding distribution and transition probabilities shown in the following figure.



By comparison with Fig. 2.2 we have:

$$h_{12}(t) = f(t)$$

$$h_{21}(t) = g(t)$$

and in addition

$$w_1(t) = h_{12}(t), \quad {}^{cc}w_1 = 1 - \int_0^t h_{12}(t) dt$$

$$N = 2$$

$$P = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

Then by simple manipulation we find,

for example, $i = 1, j = 1$,

$$\phi_{11}(s) = \frac{{}^{cc}w_1(s)}{1 - h_{12}(s) h_{21}(s)} = \frac{1 - h_{12}(s)}{s[1 - h_{12}(s) h_{21}(s)]}$$

replacing $h_{12}(s)$ and $h_{21}(s)$ with $F^*(s)$ and $G^*(s)$ we get

$$\phi_{11}(s) = \frac{1 - F^*(s)}{s[1 - F^*(s) G^*(s)]}$$

which is the same as equation (2.16). Hence by using equation (2.24) we can generate our probability function and find the respective availability. For further examples we refer the reader to Howard (1972).

2.3.3 Flow Graph and Mason's Rule

When dealing with a set of algebraic equations such as:

$$\begin{aligned}
 a_{11}X_1 + a_{12}X_2 + a_{13}X_3 + a_{14}X_4 &= b_1 \\
 a_{21}X_1 + a_{22}X_2 + a_{23}X_3 + a_{24}X_4 &= b_2 \\
 a_{31}X_1 + a_{32}X_2 + a_{33}X_3 + a_{34}X_4 &= b_3 \\
 a_{41}X_1 + a_{42}X_2 + a_{43}X_3 + a_{44}X_4 &= b_4
 \end{aligned}
 \tag{2.26}$$

The a's may be regarded as system coefficients, the b's as input and x's as output. If the equations are solvable, they can be rearranged such that one output variable with a coefficient of unity is separated from each equation. In set (2.26) above let us solve the first for X_1 , the second for X_2 etc.

$$\begin{aligned}
 X_1 &= \frac{1}{a_{11}} (b_1 - a_{12}X_2 - a_{13}X_3 - a_{14}X_4) \\
 X_2 &= \frac{1}{a_{22}} (b_2 - a_{21}X_1 - a_{23}X_3 - a_{24}X_4) \\
 X_3 &= \frac{1}{a_{33}} (b_3 - a_{31}X_1 - a_{32}X_2 - a_{34}X_4) \\
 X_4 &= \frac{1}{a_{44}} (b_4 - a_{41}X_1 - a_{42}X_2 - a_{43}X_3)
 \end{aligned}
 \tag{2.27}$$

Figures 2.4 and 2.5 show two different flow graph arrangements equivalence of Eqn. (2.27). As we can see, both flow graphs are cluttered to the point of incomprehension. These types of flow graphs are typical when one uses the Markovian model to calculate the system reliability. Therefore, if the systems

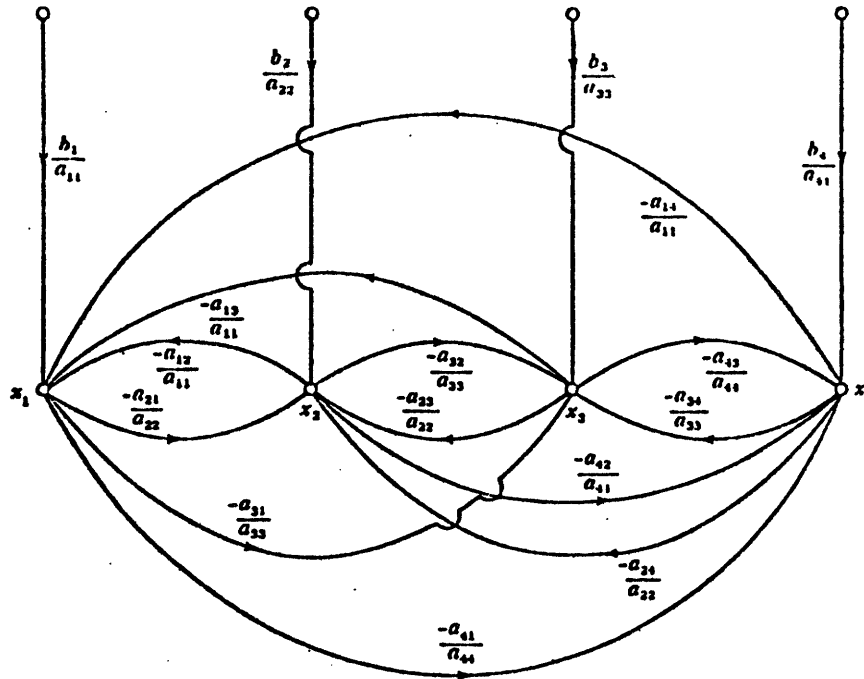


Fig. 2.4: Flow graph of Eqn. (2.27), (Henley and Williams (1973)).

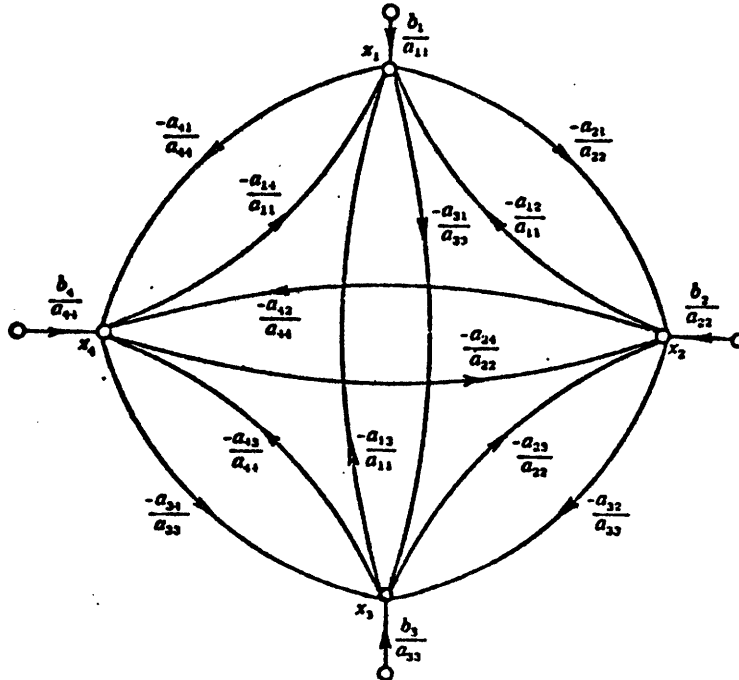


Fig. 2.5: Flow graph of Eqn. (2.27) (Henley and Williams (1973)).

are large, it is difficult to extract information from graphs. Two potential solutions to this problem are (1) eliminate the graphs and use their Matrix equivalence, or (2) reduce the graphs by combining arcs and nodes (i.e., transition and states). Here we discuss the flow graph reduction technique.

Flow graph reduction is a technique whereby a complex graph is reduced to a simpler, yet equivalent, form called a residual graph. The reduction process involves the application of a set of rules by which nodes are absorbed and branches are combined to form new branches having different gains (i.e., different transition probabilities). By repeatedly applying the reduction rules, a complex flow graph will be reduced to one having the degree of complexity desired. If a graph is completely reduced, the result is a residual graph in which one input node is connected to a selected output node by a single link. Thus, graph reduction can be viewed as a way of solving sets of linear equations.

The technique of reducing graphs is generally applicable only in the case of flow graphs where a limited number of output nodes (i.e., probabilities of state occupations) are of interest. During the reduction process, some of the nodes are necessarily absorbed and thereby lose their identity. This is a serious handicap in applications where the preservation of intermediate node is of importance. Overcoming this handicap requires that the graph be repeatedly reduced, preserving a set of selected nodes each time.

In flow graph language, the value of an unknown in a linear system can be expressed as:

$$x_i = \sum_{j=1}^n (T_{j \rightarrow i})(\text{input})_j \quad (2.28)$$

Where X_i is the i th output, n is the number of input, and $T_{j \rightarrow i}$ is the transmittance from input node j to output node i . The quantity, $T_{j \rightarrow i}$ can be thought of as the ratio of

$$T_{j \rightarrow i} = \frac{\text{contribution of input } j \text{ to output } i}{\text{input } j} \quad (2.29)$$

An ingenious technique for evaluating the transmittance T from a specified input node to an output node in a graph based on the formulation represented by equation (2.27) was devised by Mason (1957). The technique is commonly referred to as Mason's rules or "The 'Gee Whiz' Flow Graph Reduction Technique" and is expressed as:

$$T = \left(\sum_{\text{all } k} P_k \Delta_k \right) / \Delta \quad (2.30)$$

The summation in the above equation is over all of the paths leading from an input node to an output node. The quantities P_k and Δ_k are associated with the k th Path. The quantity Δ has a value equal to the value of the determinant of the system of equations (2.26).

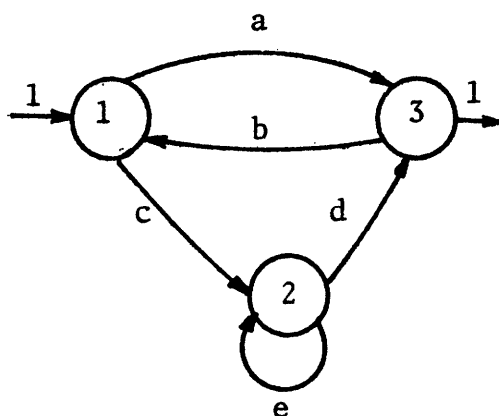
The following steps are usually useful in finding the transmittance T :

- (i) Identify "Simple Loop". A Simple Loop is a closed cycle in the graph which has no proper sub-cycle (touches no node more than once).
- (ii) Evaluate "Loop Products of Simple Loops". The loop product of a simple loop is the negative product of the transmission of arcs of the loop.
- (iii) Identify "Multiple Loops". A multiple loop is any combination of simple loops that have no nodes in common.
- (iv) Evaluate loop products of multiple loops. The loop product of a multiple loop is the product of the loop products of its simple loops.
- (v) Calculate $\Delta = 1 + \text{sum of all loop products}$.
- (vi) Identify all "paths". A path is a directed set of connected arcs with no cycle from input to the output node.
- (vii) Calculate P_k , the product of the arc transmission for each path k .
- (viii) Calculate $\Delta_k = 1 + \text{sum of the loop products of all loops sharing no node with path } k \text{ for every path}$.

(ix) Find transmittance T from Eqn. (2.30).

To clarify the above procedure, we present the following example. Let us assume we want to find the transmittance

T of the following figure:
 $1 \rightarrow 3$



Here we refer to the steps as we proceed to find T .
 $1 \rightarrow 3$

- (i) Simple loops are, 2-2, 1-3-1, 1-2-3-1.
- (ii) With loop products of $-e$, $-ab$, $-cbd$ respectively.
- (iii) The only multiple loop consists of 2-2 and 1-3-1.
- (iv) With loop product of eab , then
- (v) $\Delta = 1 - e - ab - cdb + eab$
- (vi) The paths from 1 to 3 are: 1-3 and 1-2-3
- (vii) With $P_1 = a$, $P_2 = cd$ respectively
- (viii) Then we calculate $\Delta_1 = 1 - 3$, and $\Delta_2 = 1$ to find
- (ix) $T_{1 \rightarrow 3} = \frac{a(1-e) + cd}{1 - e - ab - cdb + eab}$

Having discussed Mason's rule in flow graph reduction, we can relate Eqn. (2.25) to a flow graph. Equation (2.25) has a flow graph interpretation. In order to make it more realizable, let us rewrite Eqn. (2.25b) in the following form:

$$\Phi(s) [{}^{cc}W(s)]^{-1} [I - P \square H(s)] = I \quad (2.31)$$

Then we can write

$$\sum_{k=1}^N \frac{\phi_{ik}}{cc_{w_k}} [\delta_{kj} - p_{kj} h_{kj}(s)] = \delta_{ij} \quad 1 \leq i, j \leq n$$

and by proper rearrangement we get

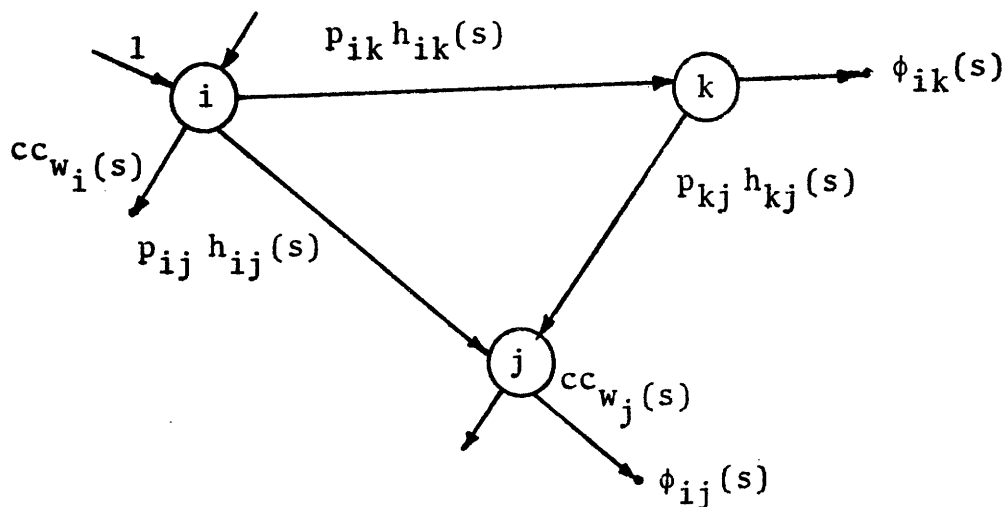
$$\frac{\phi_{ij}(s)}{cc_{w_i}(s)} = \delta_{ij} + \sum_{k=1}^N \frac{\phi_{ik}(s)}{cc_{w_k}(s)} p_{kj} h_{kj}(s) \quad (2.32)$$

for fixed i , the above equation defines N linear equations with N unknowns. If we interpret $\phi_{ij}(s)/{}^{cc}w_j(s)$ as the signal at node j and assign transmission $p_{ij} h_{ij}(s)$ to the arc from node i to node j , then $\phi_{ij}(s)$ is the product of ${}^{cc}w_j(s)$ and effective transmission node i to node j , see Fig. 2.6a.

Figure 2.6b shows the partial flow graph for fixed i , as we can see for each node j we have an output tap with transmission ${}^{cc}w_j(s)$. Finally it is worth mentioning that $\phi_{ij}(s)/{}^{cc}w_j(s)$ and $T_{i \rightarrow j}$ have the same definition (See Section 3.3.1).

$$\Phi(s) = \frac{1}{1 - P_{ii}H(s)} cc_W(s) = [I - P_{ii}H(s)]^{-1} cc_W(s)$$

- a. Matrix flow graph for the continuous time Semi-Markov process.



- b. Partial flow graph for the continuous time Semi-Markov process.

Fig. 2.6: Flow graph representation for the continuous time Semi-Markovian process.

2.4 Reliability Model of Periodically Maintained Systems

In Section 2.1, we discussed that one way to enhance the reliability (availability) of a system is to increase its redundancy. Stand-by systems are another form of redundancy. They are often added to a system of critical importance so as to ensure the safe operation of the main plant. For example, such stand-by systems are widely used in the reactors as emergency safety systems. These systems are rarely called upon to operate, but their availabilities are critical. Therefore, each system is monitored periodically in order to maximize its availability.

This class of systems cannot be analyzed by any of the previously discussed models because the failure cannot be detected when it actually occurs and because repair is not performed in a random manner; however, repair is allowed when the failure is detected. Vesely and Goldberg (1977), and Caldarola (1977) have developed two different techniques for handling this class of components. Caldarola assumes that a component undergoing periodic testing is repaired in a constant interval and becomes as good as new, while Vesely and Goldberg use the same repair policy, but they do not strictly enforce the 'as good as new' model after repair. In this study a generalized formulation is developed that accommodates different repair policies.

Figure 2.7 shows a typical time dependent unreliability of a periodically tested component with a constant repair time of

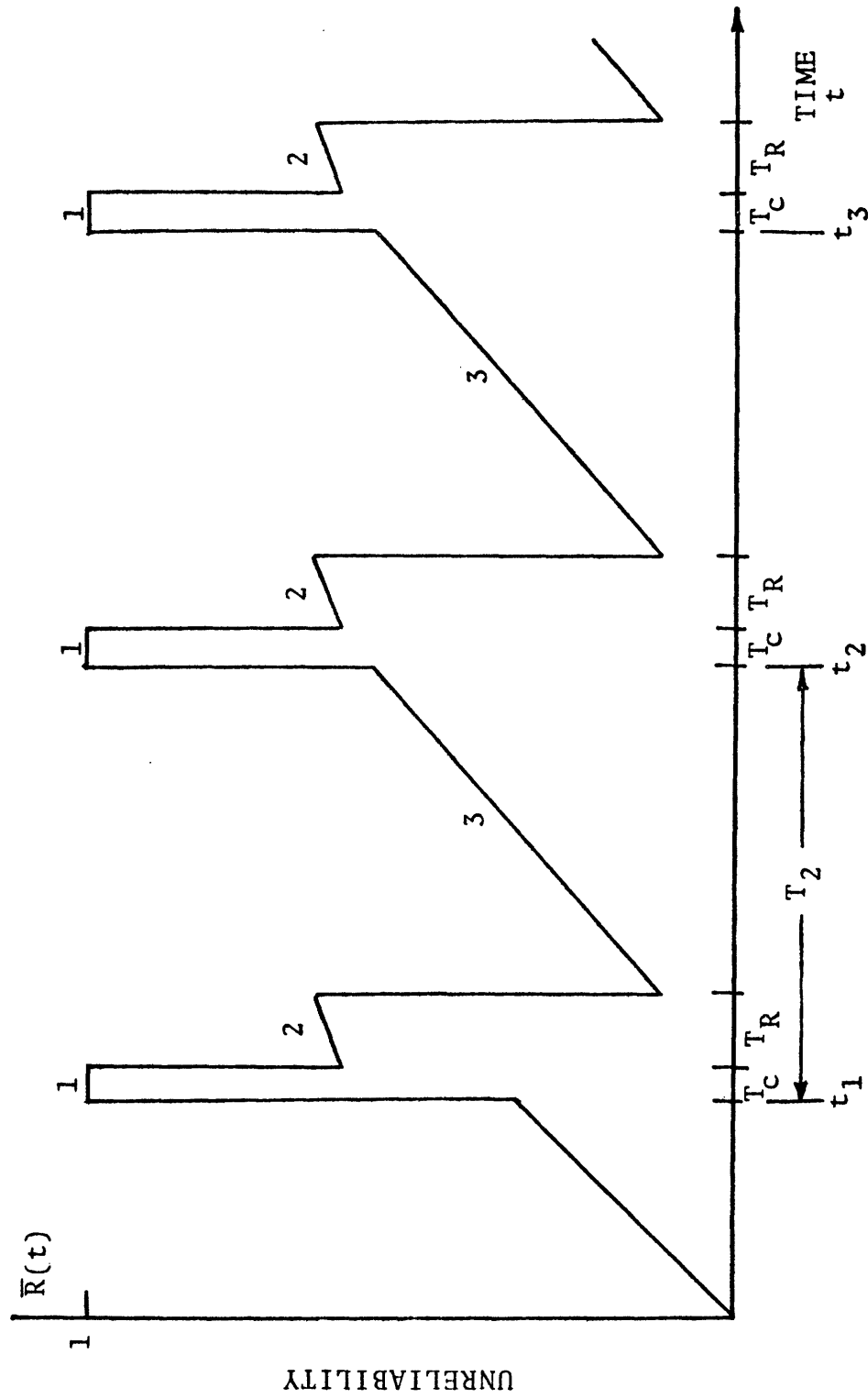


Fig. 2.7: A typical time dependent unreliability of a periodically tested component.

T_R . As we can see, the test interval is divided into three periods:

- 1) test period duration, T_C (hours),

$$t_n \leq t \leq t_n + T_C$$

- 2) repair period duration, T_R (hours),

$$t_n + T_C \leq t \leq t_n + T_C + T_R$$

- 3) between the test (stand-by) period duration, $T_2 - T_C - T_R$ (hours),

$$t_n + T_C + T_R \leq t \leq t_{n+1}$$

In order to predict the unreliability of a periodically tested component one has to account for the following contributions:

A) Test period contribution

The system may be unavailable during the inspection period for either of the following causes:

- 1) The system is unable to override the test with probability q_0 should an emergency signal require its operation.
- 2) System has failed before the test with probability Q .

Therefore, the unreliability contribution during this period will be:

$$\bar{R}(t) = Q + (1 - Q)q_0; \quad t_n \leq t \leq t_n + T_C$$

where

$$Q = \bar{R}(t) \Big|_{t = t_n} \quad (2.33)$$

B) Repair and between the test (stand-by) period contributions, $t_n + T_c < t \leq t_{n+1}$

Since we are interested in a general repair policy, we cannot differentiate between the repair and stand-by period contributions as we did for our typical example. Here, we consider a general repair time distribution function, $g(t)$, for the repair policy.

The system is unavailable at time t after the test if:

- 1) The system was up at time $\underline{0}$ (at the end of test time) and it failed at $\underline{t}$ randomly according to failure time density function $f(t)$; or
- 2) The system was down at time $\underline{0}$ and repaired with the following possibilities:

- a) it is still under repair = $1 - G(t)$
- b) it was repaired at time τ , ($0 < \tau < t$), but failed again at t with probability =

$$\int_0^t G(\tau) f(t - \tau) d\tau$$

NOTE: in part b) we assume the system will be repaired once per interval.

Therefore we can write:

$$\bar{R}(t) = \alpha F(t) + \beta [1 - G(t) + \int_0^t G(\tau) f(t-\tau) d\tau] \quad (2.34)$$

Where

α = the probability that the system was up after the test (reference time: $t = 0$)

β = $1 - \alpha$ = the probability that system is down at time zero

$$F(t) = \int_0^t f(t) dt$$

$$G(t) = \int_0^t g(t) dt$$

In the above equation β is the sum of two different probabilities that have happened prior to the start of this cycle. In other words, if the above equation is written for the time interval t_n to t_{n+1} , the value of β is related to the unreliability outcome of t_{n-1} to t_n . Hence, for generalizing the Eqn. (2.34) we start at the first test interval $t = t_1$ where the unreliability is:

$$\bar{R}_1(t) = F(t)$$

and

$$Q_1 = \bar{R}_1(t) \Big|_{t=t_1} = F(t_1) = F_1 \quad (2.35)$$

The system will be down after the first test with probability β_1 because either it has failed during the test or failed prior to test. Therefore:

$$\beta_1 = P_f + (1 - P_f)Q_1 = P_f + \bar{P}_f Q_1 \quad (2.35a)$$

where:

P_f = probability that the test caused the failure.

Consequently, by writing the successive Q_i and β_i we can write

$$\begin{aligned} Q_2 = (1 - \beta_1)F + \beta_1 \int_0^{T_2 - T_c} G(\tau)f(t - \tau)d\tau \\ + \beta_1[1 - G(T_2 - T_c)] \end{aligned} \quad (2.35b)$$

However, the third term of the above equation is almost zero because T_2 , in practical cases, is much larger than the average repair time, and if we replace the integral term by $\underline{Y}$ we have:

$$Q_2 = (1 - \beta_1)F + \beta_1 Y$$

Substituting for β_1 from Eqn. (2.35a), and rearranging the above equation, we get:

$$Q_2 = X - Q_1[X - \bar{P}_f Y] + P_f Y \quad (2.35c)$$

$$\beta_2 = P_f + \bar{P}_f Q_2$$

$$Q_3 = X - X[X - \bar{P}_f Y] + Q_1[X - \bar{P}_f Y]^2 - P_f Y[X - \bar{P}_f Y] + P_f Y$$

$$\cdot \quad \cdot \quad \cdot$$

$$\cdot \quad \cdot \quad \cdot \quad \frac{1 - (-1)^{N-1}[X - \bar{P}_f Y]^{N-1}}{1 + X - \bar{P}_f Y}$$

$$Q_N = (X + P_f Y) + (-1)^{N-1} Q_1[X - \bar{P}_f Y]^{N-1} \quad (2.36)$$

$$\beta_N = P_f + \bar{P}_f Q_N \quad (2.36a)$$

Where:

$$F = F(t) \Big|_{t = T_2 - T_c}$$

$$X = \bar{P}_f F$$

N = number of cycle

The general unreliability equation for time interval t_n to t_{n+1} is:

$$\bar{R}_n(t) = (1 - \beta_{n-1})F(t) + \beta_{n-1}[1 - G(t) + \int_0^t G(\tau)f(t-\tau)d\tau]$$

$$0 \leq t \leq T_2 - T_c \quad (2.37)$$

Usually the quantity, $[X - \bar{P}_f Y]$ is very small and the value of Q_N quickly reaches its asymptotic value which is given by:

$$Q_\infty = \frac{X + P_f Y}{1 + X - \bar{P}_f Y} \quad (2.38)$$

The asymptotic unreliability equation will then read:

$$\bar{R}_\infty(t) = F(t) + [\bar{P}_f \frac{X + P_f Y}{1 + X - \bar{P}_f Y} + P_f]$$

$$[1 - G(t) - F(t) + \int_0^t G(\tau)f(t - \tau)d\tau]$$

$$0 \leq t \leq T_2 - T_c \quad (2.39)$$

To show the applicability of the above equation, we give the following examples and compare the results with Vesely et al. (1977) and Caldarola (1977) for constant repair and exponential failure distributions.

First, let us show the above referenced equations (for detailed information see the references):

a) Vesely et al. (1977). For the case where

$$P_f = 0 \text{ and } q_0 = 1$$

$$1) \bar{R}(t) = \lambda(t - T_c) \quad T_R + T_c < t < T_2 \quad (2.40)$$

$$2) \bar{R}(t) = \lambda(T_2 - T_c) + (1 - \lambda(T_2 - T_c)) \frac{1}{2} \lambda T_R$$

$$T_c < t \leq T_R + T_c \quad (2.41)$$

$$3) \bar{R}(t) = 1 \quad 0 < t \leq T_c$$

b) Caldarola (1977).

$$\begin{aligned} 1) \bar{R}_\infty(t) &= 1(t) - 1(t - T_c) + F(t - T_c) \\ &+ \frac{F(T_2 - T_c)}{1 + F(T_2 - T_c) - F(T_2 - T_c - T_R)} \\ &[1(t - T_c) - 1(t - T_c - T_R) - F(t - T_c) \\ &+ F(t - T_c - T_R)] \\ &0 \leq t \leq T_2 \quad (2.42) \end{aligned}$$

2) Approximated equation used in PL-MODT (Modarres (1979)). (2.43)

$$R(t) \Big|_{\text{approx}} = 1 - \exp(-\lambda_{\text{eff}} t) [1 - \exp[-(\frac{t}{T_c})^q]]$$

where

$$\lambda_{\text{eff}} = \lambda \frac{T_2 - T_c}{T_2} \left(\frac{T_2 - T_c}{T_2} + \frac{2T_R}{T_2} \right) + 2 \frac{T_c}{T_2} \left(1 - \frac{\Gamma(\frac{1}{q})}{\Gamma(q)} \right)$$

$$q = \ln(3 - \ln(\lambda T_c))$$

$\Gamma(\cdot)$ = Gamma function

The significance of q is: it must be large enough to cause $\bar{R}_\infty(t) \big|_{\text{approx}}$ to become:

$$\bar{R}_\infty(t) \big|_{\text{approx}} = 1 - \exp(-\lambda_{\text{eff}} t) \quad \text{for } t > T_c$$

Now, let us find our equation for constant repair and exponential failure distribution.

$$G(t) = \begin{cases} 0 & t \leq T_R \\ 1 & t > T_R \end{cases}$$

$$f(t) = \lambda e^{-\lambda t}, \lambda \text{ is failure rate.}$$

For consistency with the Eqn. (2.42), we change the reference time in Eqn. (2.39) to the start of the cycle:

$$\begin{aligned} \bar{R}_{\infty}(t - T_c) = & (1 - \beta_{\infty})F(t - T_c) + \beta_{\infty}[1 - G(t - T_c) \\ & + \int_0^{t-T_c} G(\tau)f(t - T_c - \tau)d\tau] \end{aligned} \quad (2.44)$$

By replacing $G(\cdot)$, β_{∞} , and taking into account that $P_f = 0$ and

$$\begin{aligned} Y &= \int_0^{t-T_c} G(\tau)f(t - T_c - \tau)d\tau \\ &= \int_0^{T_2-T_c} f(t - \tau')d\tau' = F(T_2 - T_c - T_R) \end{aligned}$$

where $\tau' = T_c + \tau$

Equation (2.44) will read:

$$\begin{aligned} \bar{R}_{\infty}(t - T_c) = & F(t - T_c) + \frac{F(T_2 - T_c)}{1 + F(T_2 - T_c) - F(T_2 - T_c - T_R)} \\ & [1 - G(t - T_c) - F(t - T_c) + F(t - T_c - T_R)] \end{aligned} \quad (2.45)$$

In the cases where $[F(T_2 - T_c) - F(T_2 - T_c - T_R)]$ is small we can find an approximate formulation from the above equation for t in the intervals of $[T_c, T_c + T_R]$ and $[T_c + T_R, T_2]$, where T_R is average repair time.

For $T_c < t < T_c + T_R$ we can write:

$$\bar{R}_\infty(t) \Big|_{\text{approx}} = F(t - T_c) + F(T_2 - T_c)[1 - F(t - T_c)] \quad (2.46)$$

Because $G(t - T_R)$, $F(t - T_c - T_R)$ are both zero when their arguments are less than or equal to zero, then we can find the average unreliability over T_R .

$$\begin{aligned} \bar{R}_\infty \Big|_{\text{avg}} &= \frac{1}{T_R} [(1 - F(T_2 - T_c)) \int_0^{T_R} F(t) dt \\ &\quad + \int_0^{T_R} F(T_2 - T_c) dt] \end{aligned}$$

Replacing for $F(\cdot)$ and approximating the exponential term (i.e., $e^{-x} = 1 - x + \frac{x^2}{2}$ where $x < 0.01$) we get

$$\bar{R}_\infty \Big|_{\text{avg}} = [1 - F(T_2 - T_c)] \frac{1}{2} \lambda T_R + F(T_2 - T_c) \quad (2.47)$$

For $T_c + T_R < t < T_2$ we can write:

$$\begin{aligned} \bar{R}_\infty(t) \Big|_{\text{approx}} &= F(t - T_c) + F(T_2 - T_c)[F(t - T_c - T_R) \\ &\quad - F(t - T_c)] \end{aligned} \quad (2.48)$$

But since we assumed that the second term is small compared to the first term, because the bracket term is very

small, Eqn. (2.41) will be reduced to:

$$\bar{R}_{\infty}(t) \Big|_{\text{approx}} = F(t - T_c) \quad T_R + T_c < t < T_2 \quad (2.49)$$

The application of Eqns. (2.39 to 2.43), (2.45), (2.47) and (2.49) to a typical system with $\lambda = 10^{-3}/\text{hr}$, $T_R = 21 \text{ hrs.}$, $T_2 = 720 \text{ hrs.}$, $T_c = 1.5 \text{ hrs.}$ is shown in Table 2.1 and Fig. 2.8. As we can see, Caldarola's first equation and Eqn. (2.45) of this study give exactly the same results (they are basically identical equations). Our approximated equations, i.e., Eqns. (2.47), (2.49) predict very well in comparison to the exact results (compare columns 4 and 5). Both the equations used in PL-MODT (cf. Eqn. (2.43)) underestimate the unreliability during the repair and overestimate at the end of the interval. Vesely's equations overestimate conservatively everywhere because of the nature of the approximation used in the formulation. In fact, if one keeps the exponential terms instead of its first two terms approximation in Vesely's model, one will get the same equation as we obtained by approximating our general equation (cf. Eqns. (2.47), (2.49)). We should mention that the approximation is only true if $[F(T_2 - T_c) - F(T_2 - T_c - T_R)] \ll 1$.

In another example, we consider the following repair distributions:

- 1) exponential with $\mu = \frac{1}{T_R}$, $T_R = 21 \text{ hrs.}$

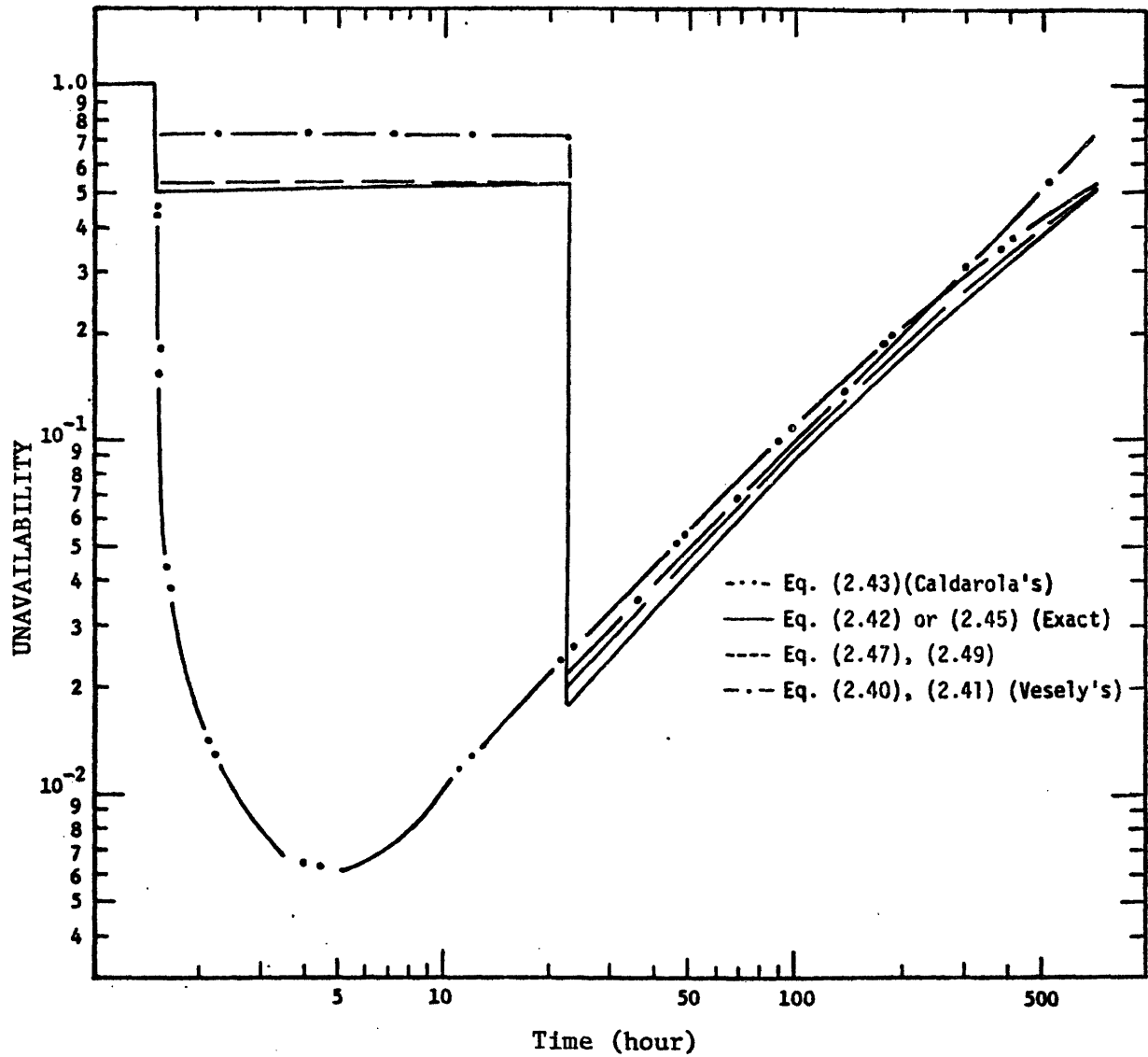


Fig. 2.8: Comparison of the unavailabilities as calculated from the equations developed in this study and those of Vesely's and Caldarola's for a periodically tested component.

Table 2.1: COMPARISON OF MODEL DEVELOPED IN THIS STUDY
FOR PERIODICALLY MAINTAINED COMPONENT WITH VESELY'S AND CALDAROLA'S MODEL
FOR A TYPICAL EXAMPLE

Time Hour	CALDAROLA		THIS STUDY		Vesely Eqns. (2.40), (2.41)
	Eqn. (2.43) Use in PLMODT	Eqn. (2.42) Exact	Eqn. (2.45) Exact	Eqns. (2.47), (2.49) Approx.	
0.0	1	1	1	1	1
1.5 ⁺	0.368871	0.5072692	0.5072692	0.5176356	0.721456
11.0	0.0115276	0.5119280	0.5119280	0.5176356	0.721456
21.0	0.0218918	0.5167043	0.5167043	0.5176356	0.721456
22.5 ⁻	0.0234371	0.5175086	0.5175086	0.5176356	0.721456
22.5 ⁺	0.0234371	0.0180854	0.0180854	0.020781	0.021000
100.0	0.1000	0.084042	0.084042	0.0938043	0.09850
200.0	0.1900	0.171207	0.171207	0.1800	0.19850
400.0	0.3440	0.321441	0.321441	0.328674	0.39850
600.0	0.4687	0.4444	0.4444	0.450364	0.59850
720.0	0.531826	0.5072692	0.5072692	0.512517	0.71850
Average Unavail.	0.2624	0.2965	0.2965	0.3033	0.3831

- 2) 2nd order Erlangian $\mu = \frac{2}{T_R}$, $T_R = 21$ hrs.
- 3) lognormal with $\mu = \ln(T_R)$, $\sigma = 0.5$, $T_R = 21$ hrs.

Table 2.2 summarizes the unreliability results by utilizing Eqn. (2.45). None of the above references are able to predict the unreliabilities for these distributions. Therefore, a comparison was not possible. However, a comparison of the aforementioned distributions with their given properties shows a small difference in unreliability outcome for this typical example, which signifies that we can interchange distributions as long as we maintain their mean values nearly identical.

In the above example, both exponential and 2nd order Erlangian distribution have the same mean, i.e., $T_R = 21$, and the lognormal has a median of 21 hours. In view of the complexity of Eqn. (2.45) for the above distributions, the exponential distribution is a good candidate for repair, i.e., it is simple and gives conservative values (see Fig. 2.9). For detailed equations used in the analysis of Table 2.2, see Appendix A.1.

Table 2.2: APPLICATION OF EQN. (2.45) TO DIFFERENT REPAIR DISTRIBUTIONS

Time Hours	Exponential $\mu = \frac{1}{21}$	Erlangian $\mu = \frac{2}{21}$	Lognormal $\mu = \ln(21),$ $\sigma = 0.5$	
0.0	1	1	1	
1.5	0.5072133	0.5072414	0.5070952	
11.0	0.328243	0.3960357	0.462927	
21.0	0.213350	0.238204	0.264678	
22.5	0.200723	0.219048	0.237763	
50.0	0.088429	0.065494	0.051507	
75.0	0.07630	0.06455	0.051295	
100.0	0.08870	0.084446	0.066967	
150.0	0.129061	0.128677	0.1022454	
200.0	0.171160	0.171167	0.137643	
400.0	0.32137	0.32140	0.277765	
600.0	0.4444	0.4444	0.41945	
720.0	0.5072133	0.5072414	0.5070952	
	0.295537	0.295498	0.2705288	Unavailability Per Demand

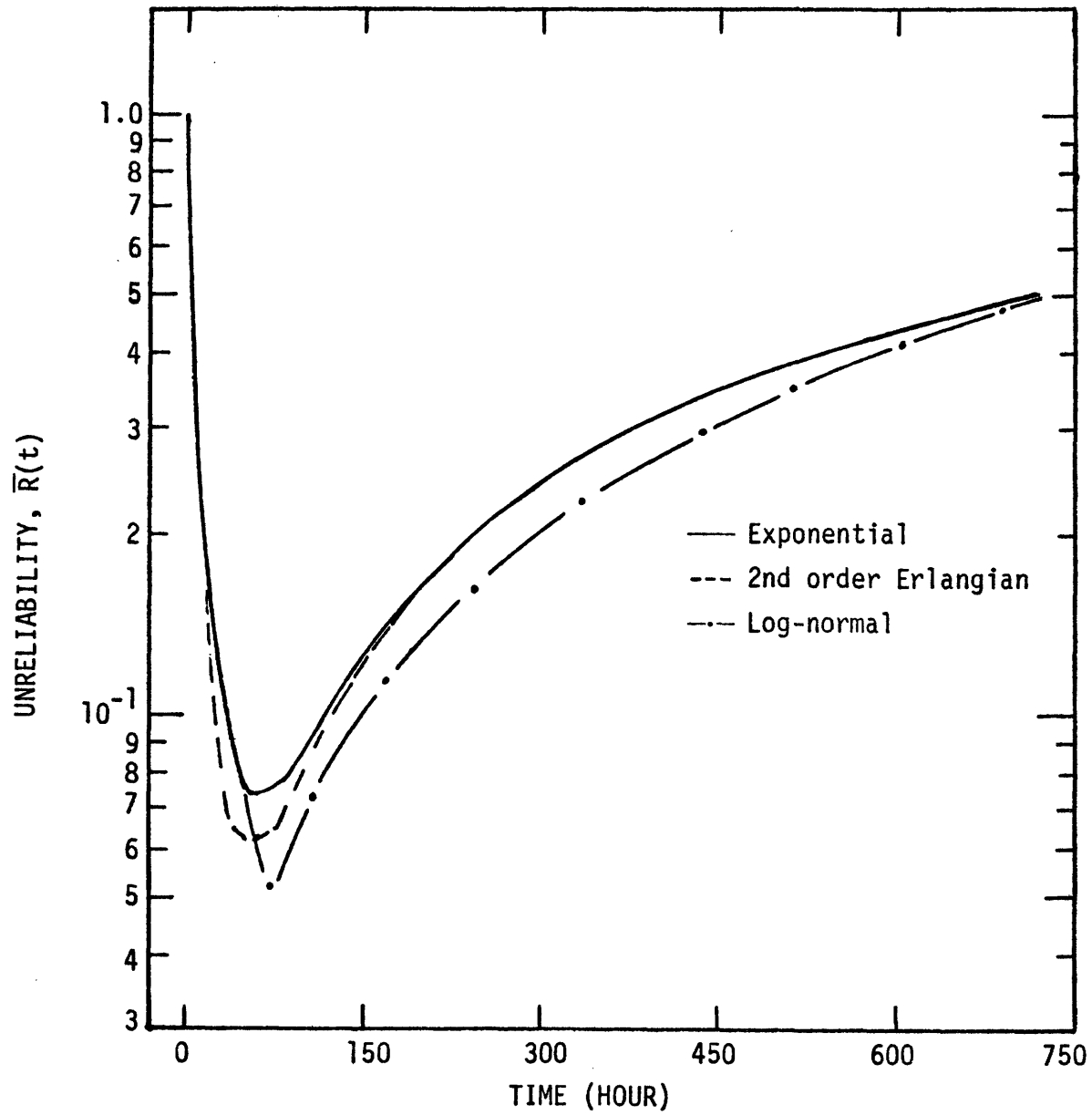


Fig. 2.9: Comparison of the application of Eqn. (2.45) to the different repair distributions.

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

CHAPTER 3: PRINCIPLE STRUCTURE OF THE CODE UNRAC

3.1 Introduction

The knowledge of the availability of the emergency safety features in a reactor is of major importance in assessing reactor safety. One method which has been used successfully in the past and is being reiterated in the wake of Three Mile Island is Fault Tree Analysis (FTA). Most vendors and utilities have been instructed by the Nuclear Regulatory Commission (NRC) to evaluate the availability of their emergency systems to ensure a safe condition of the reactor should an accident occur. Because of the complexity of the system, one has to have a tool (generally a computer code), to predict the availability of the system. To do so, there exist several codes (cf. Chapter 1), but each has its own restrictions and usually one requires two or more codes to do a complete job. Therefore, one has to be familiar with three or more codes in order to adequately predict the required system availability. In such cases, the probability of making errors in data preparation will increase greatly and the job will become cumbersome. Hence, there is an incentive to develop a code which is comparatively easy to use and, in addition, can accommodate the complete job in one step. In this study, we develop a computer package which does the following:

- 1) generates the minimal cut sets;
- 2) finds the importance of each basic component relative to system failure;
- 3) evaluates the point estimate, time dependent and average unavailability and/or unreliability of the system under study; and
- 4) simulates the top event distribution by Monte-Carlo and finds the approximate distribution which represents the top event by the method of moment matching.

The package can be used for carrying out any of the above analyses individually or totally. To discuss the basic principle of the code, we divide the package into three sections:

a) Cut set generator

This part of the code generates the cut sets and finds the minimal cut sets which are essential for evaluating the top event unavailability.

b) Time dependent unavailability evaluator

This part of the code calculates basic component unavailability depending upon the nature of the components (maintained or non-maintained) and parameters (failure and repair rates) involved.

c) Monte-Carlo simulator

This part of the code simulates each component's distribution in order to find the top events

distribution and to determine the uncertainty in the evaluation of top event from the given spread of the component's parameters

3.2 Cut Set Generator

3.2.1 Introduction

The first step in FTA is to find the minimal cut sets. In the past decade, several codes have been developed by different researchers. The most popular among these are PREP, by Vesely and Narum (1970), MOCUS by Fussel et al. (1974), TREEL-MICSUP by Pande et al. (1975), SETS by Worrel (1975), and WAM-CUT by Erdmann et al. (1978). The above selection is a good example of the versatility of the techniques which have been used to generate the minimal cut sets. PREP is the first code in this series. Because of the inherent limitations of the PREP code, MOCUS was written to replace it. MOCUS uses a top down algorithm. The algorithm starts from the Top event of the fault tree and moves, by successive substitution of the gate equations, down the tree until only basic events remain in the list of possible Top-event failure modes(i.e. cut sets). In another study it has been found that by proper storage and restructure of the fault tree, one can decrease both the storage and time necessary to find the cut set relative to MOCUS (Chatterjee (1974)). TREEL-MICSUP is based on the above finding and the algorithm used in this code is a bottom up one. In this code, the tree will be restructured first and then the code starts to find the cut set and respective minimal cut sets.

In SETS, Set Equation Transformation System, the code symbolically manipulates Boolean equations formed by a set of events operated on by a particular set of union, intersection and complement operators. Here the fault tree should be input as a Boolean equation to represent each event as a function of its input events.

A Boolean equation is a mathematical representation of a gate, i.e., for two components inputted to an AND or an OR gate one has to write:

$$C = A + B \quad \text{for OR gate}$$

$$C = A * B \quad \text{for AND gate}$$

$$C = \text{is the output signal}$$

The last code mentioned above, WAM-CUT, uses the TREEL-MICSUP algorithm. First the code restructures the fault tree so that each gate can only accept two inputs. The restructured tree will be stored from low level gates -- gates with components input only -- to high level gates and finally top event. Second, the code uses binary representation of basic components, i.e., 1 for failed state and 0 for unfailed state of the component, to be able to store each component in the cut set by a bit of the word which represents the cut set. By this method, one can store as many as 63 components in a cut set word length in CDC 7600 or 31 components in IBM 370/168 as compared to 63 or 31 word lengths needed to store in MOCUS or MISCUP.

All the above mentioned codes are written for large trees and the initial storage required for each code is prohibitive, i.e., WAM-CUT needs 1440K-byte storage for any job of up to 1500 components and/or gates. Rosenthal (1975) pointed out that in general we cannot hope to find a "fast" algorithm for arbitrary fault trees, and the only hope for analyzing large trees is through the application of tree decomposition methods. Therefore, there is an incentive to look for an algorithm which requires less storage and which is comparatively fast to analyze the large decomposed trees. In addition, it should be able to couple the code with others without hampering the code performance and exceeding certain storage limits. For qualitative and quantitative analysis, one needs the top event minimal cut sets only, therefore one should be careful not to sacrifice storage and efficiency in order to get more capabilities out of a given code.

3.2.2 Cut Set Generator Used in UNRAC

To generate the minimal cut set from a given fault tree, we chose BIT and modified it to suit our needs in this study. BIT is a portion of an unpublished work by Dr. Wolf (1975) in which one could only use a fault tree with a limited number of components and/or gates. After a series of modifications and improvements, the BIT code was benchmarked against MODCUT (Modarres 1979) and WAM-CUT (EPRI-1978) to check its accuracy and efficiency. The reason for choosing these two codes was that both use binary representation of the components to

generate the minimal cut sets. However, MODCUT is written in the PL/1 computer language whereas WAM-CUT and BIT are in FORTRAN IV.

The BIT code uses a top down algorithm. It generates the cut sets by successive replacement of the gates through the fault tree logic by their respective inputs. BIT is written in such a way that the user could easily follow the steps in the code and, to prevent confusion, the major steps in the code have been separated from the main routine by proper subroutines. Figure 3.1 shows the flow chart and steps used in the BIT.

Since cut set generation is a time consuming process, a discrimination procedure which eliminates cut set of given size has been encoded to accelerate the computation when quantitative evaluation is a desired goal and the component unavailabilities are small. The discrimination process is at its best when one uses a top down algorithm. Otherwise, one has to find all the intermediate cut sets before one is able to discriminate them. In order to see how the discriminatory process is important for some fault tree see Fig. 3.2 (EPRI (1975)).

BIT is also capable of handling both complement and basic events. Therefore, it allows the user to have NAND, NOR, NOT and Exclusive OR (EOR) gates in his fault tree. However, by proper transformation of special gates, such as NAND, NOR and EOR, one can easily end up with the basic logics, i.e., AND, OR and NOT gates. Fig. 3.3 shows such transformation.

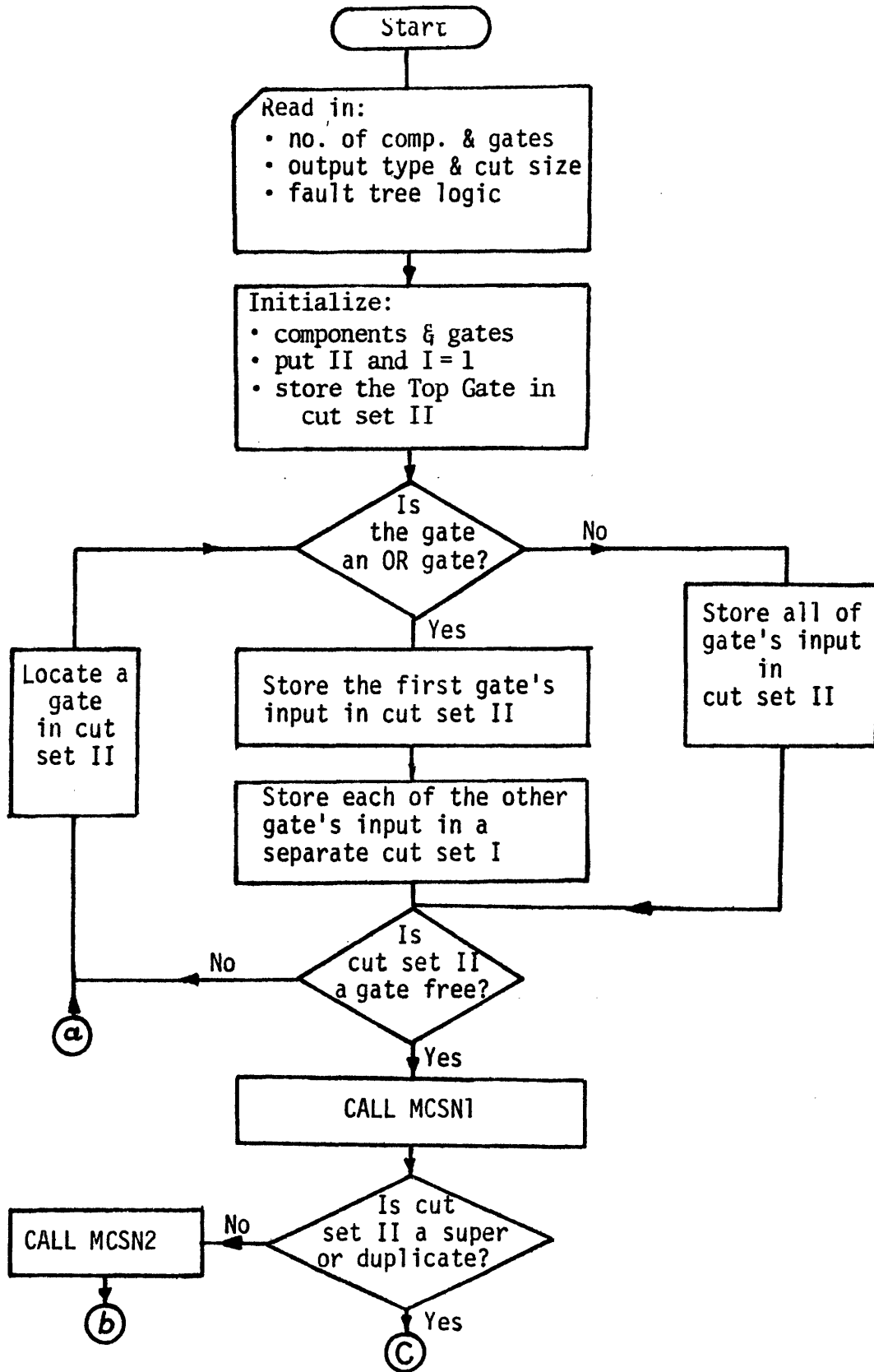


Fig. 3.1: Flow chart and steps used in cut set generator BIT.

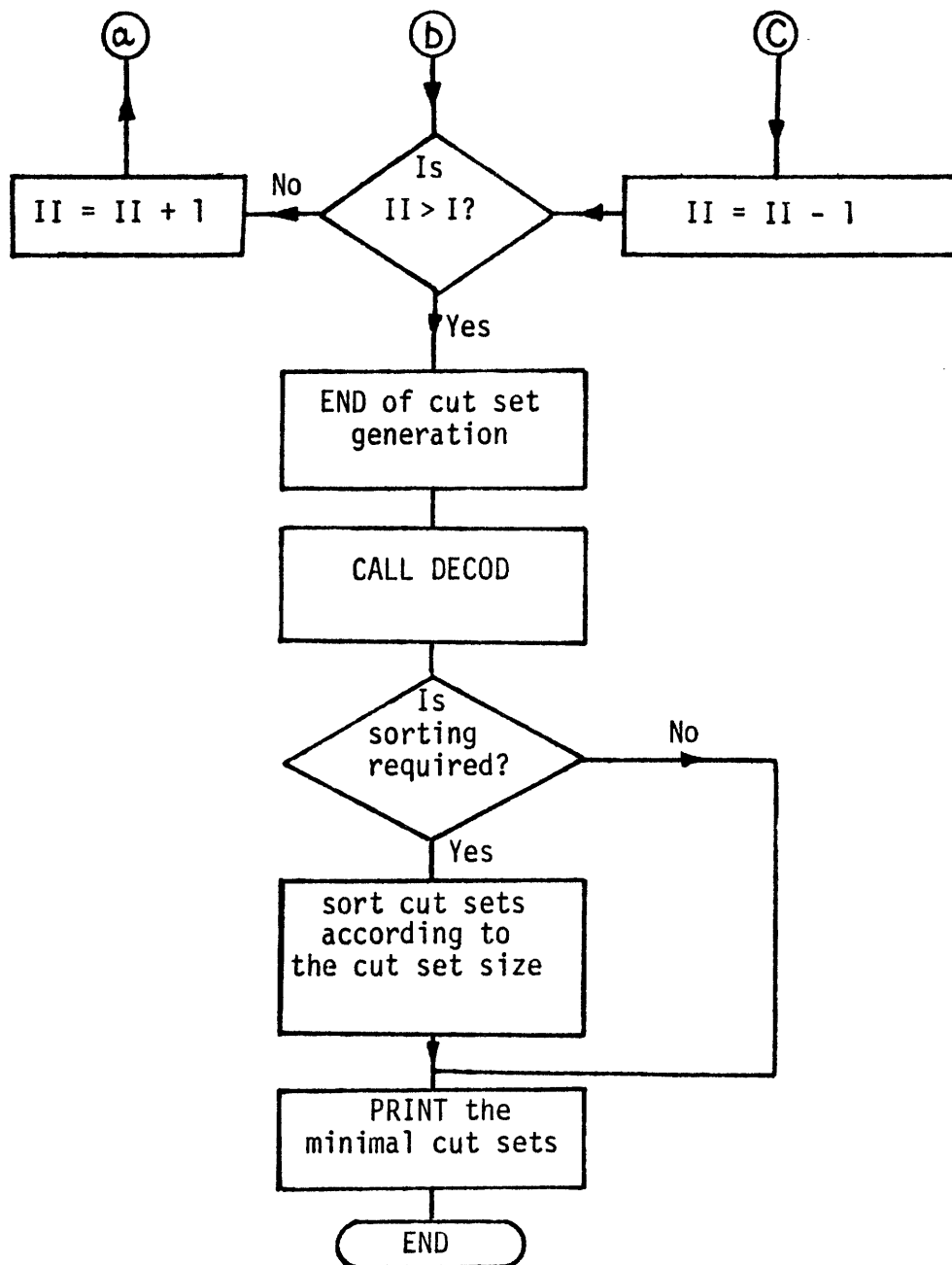


Fig. 3.1: continued.

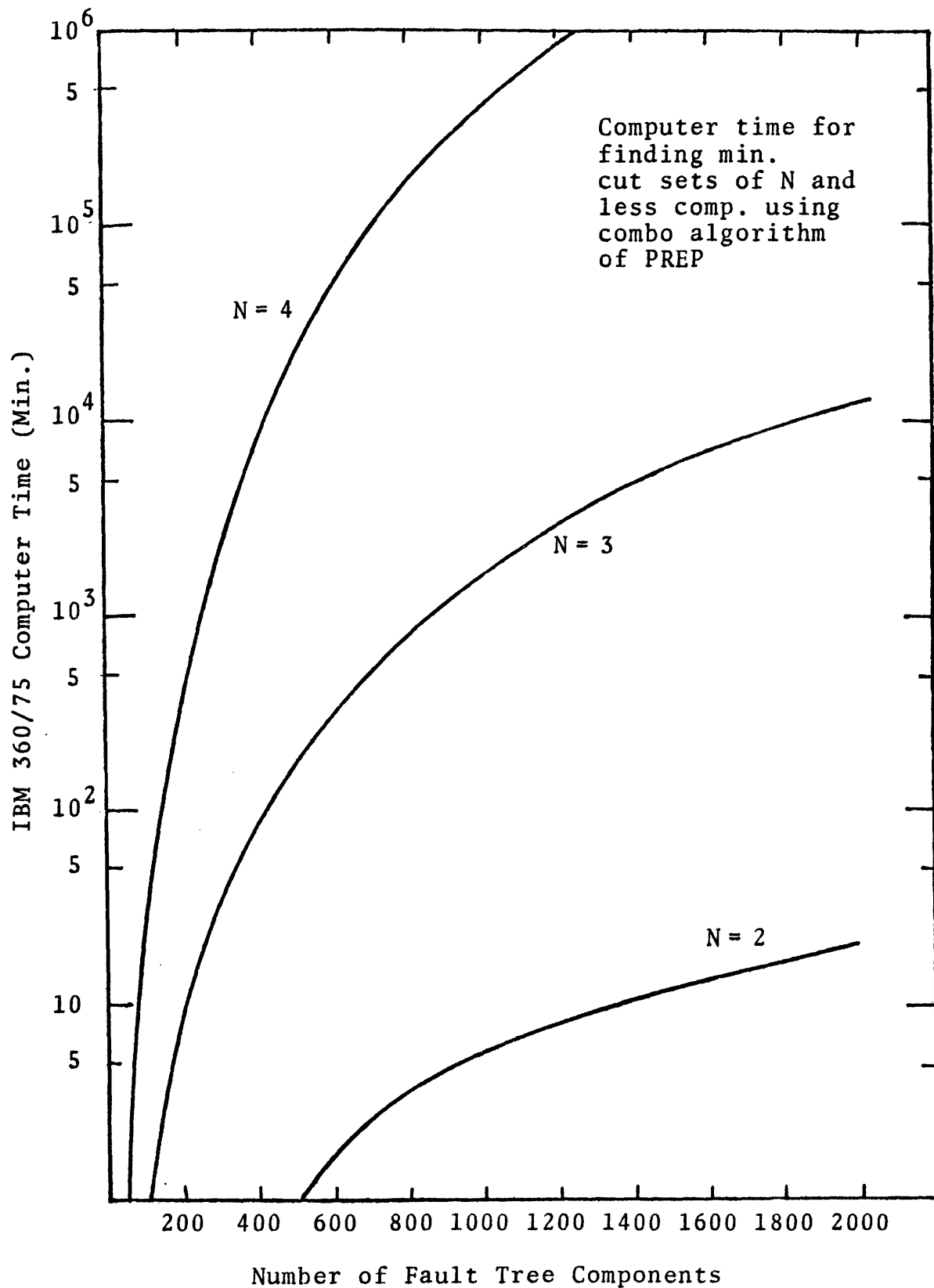


Fig. 3.2: Computer time to deterministically find minimal cut sets, (EPRI (1975))

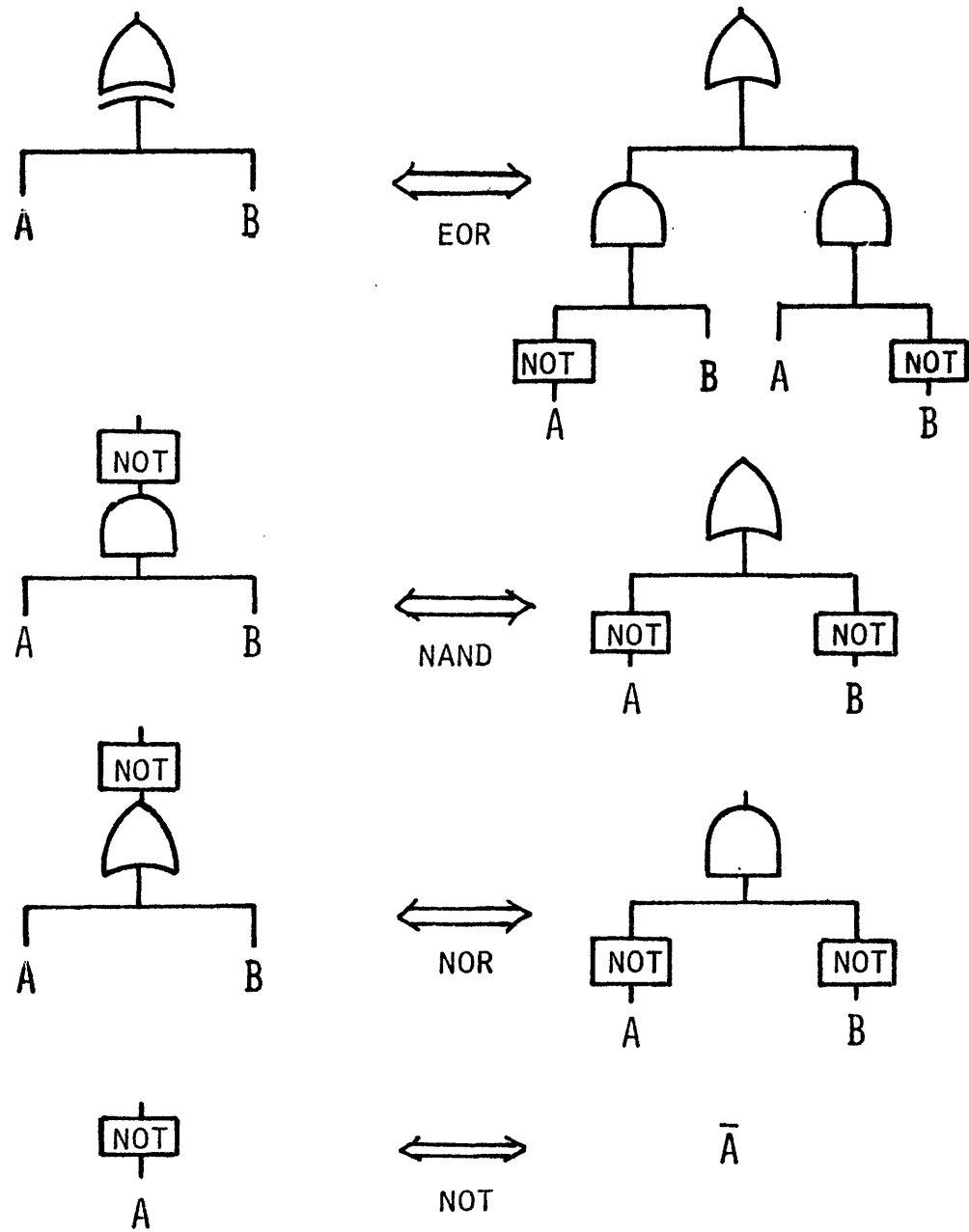


Fig. 3.3: Equivalent transformation of EOR, NAND, NOR, and NOT gates

To clarify the BIT algorithm described above, we do the following example. Figure 3.4 shows an electrical circuit which was originally used by Fussell (1973) and has been used by several authors by including and/or excluding different parts from it to show their methods of generating cut sets (prime implicant), i.e., see Bennett (1975) and Henley & Kumamoto (1978). Figure 3.5 shows the transformed fault tree after replacing EOR by basic gates and combining some of the successive gates given in Bennett (1975). By proper initialization of components and gate, we have a total of 9 components and 11 gates.

BIT starts by initializing each component and/or gate uniquely by storing 1 in a bit of the word related to the component and/or gate number, i.e., component I will be stored as:

$$ME(I) = 2^{*(I-1)} \quad I \leq 31$$

for $I = 1$

$$ME(1) = (0000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0000 \ 0001)_2$$

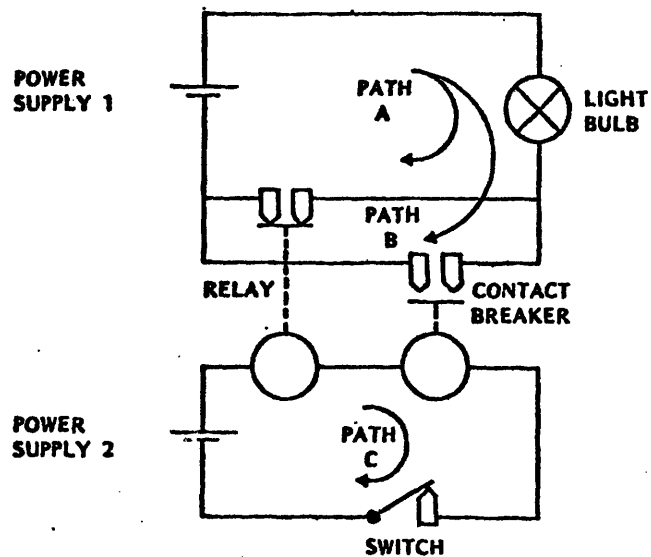
or to show it more compactly

$$ME(1) = \underline{0} \ \underline{0} \ \underline{0} \ \underline{0} \ \underline{0} \ \underline{0} \ \underline{0} \ \underline{0} \ 0001 = B'1'$$

and for $I = 20$

$$ME(20) = \underline{0} \ \underline{0} \ \underline{0} \ 1000 \ \underline{0} \ \underline{0} \ \underline{0} \ \underline{0} = B'20'$$

$B'.'$ = represents bit occupation No. of the word (The rest of bits are zero)

**TOP EVENT****PRIMARY FAILURES**

- Z : No light**
A : Bulb failure
B : Power supply 1 failure
C : Relay contacts stuck open
D : Circuit breaker stuck open
E : Switch stuck open
F : Switch stuck closed
G : Power supply 2 failure
H : Relay coil open circuits
I : Circuit breaker coil open circuits

INITIAL CONDITIONS : Switch closed, relay contacts closed, circuit breaker contacts open.

CIRCUIT ACTION : If relay contacts open, an operator opens the switch causing the contact breaker to close thereby restoring power to the bulb.

NOT-ALLOWED EVENTS : Operator failure, wiring failure, secondary failure.

Fig. 3.4: Detailed information of operation and failure modes of an electrical wiring system (Bennett (1975)).

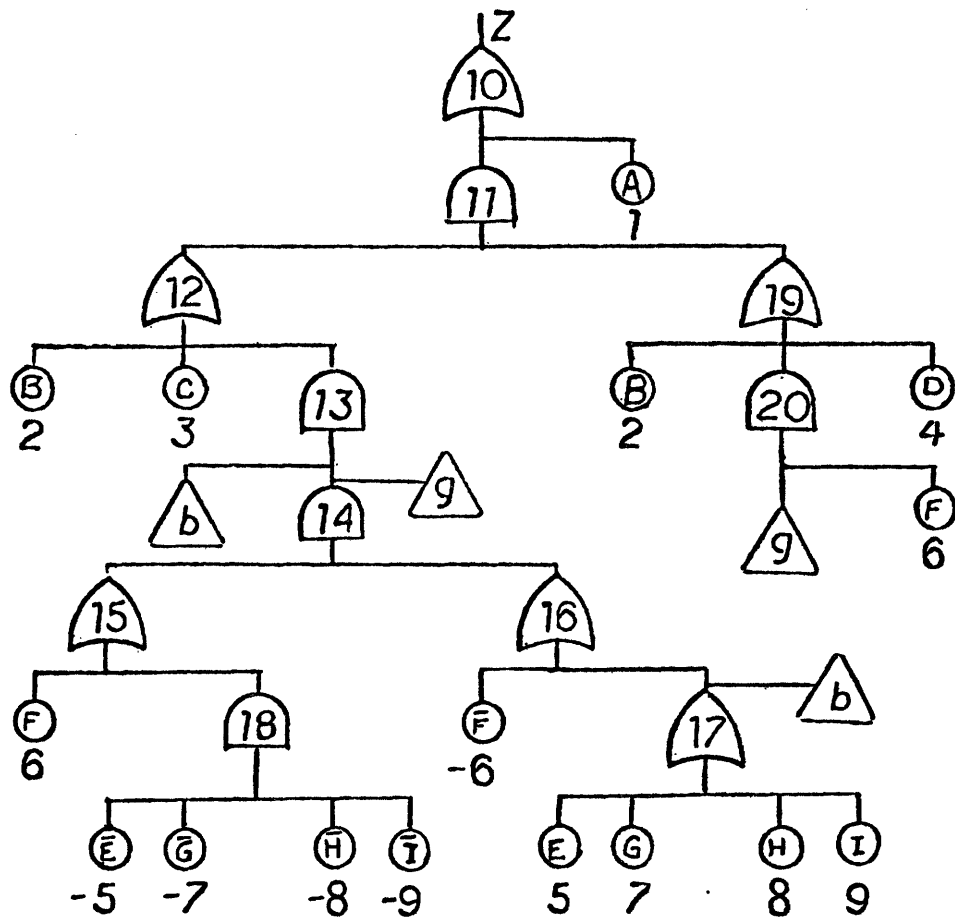


Fig. 3.5: Compact fault tree of the electrical wiring system.

In the top down algorithm we start by reading in the fault tree from the top event and for each input to the OR gate we increase the number of cut sets. Each cut set in the code consists of two parts: the positive part which stores all the basic events and gates and the negative parts in which we only store the complements of the events called KM and MK array in the code respectively. Since there are only 31 available bits in a word (IBM 370/168), we have to combine a series of word lengths to get the desired components and/or gates capability in the code. Here we use only 1 word length because the total number of components and gates is twenty.

The top gate is an OR gate, gate 10. Therefore, we have two cut sets:

$$\begin{array}{ll} \text{KM}(1) = \text{B}'\underline{11}' & \text{MK}(1) = 0 \\ \text{KM}(2) = \text{B}'1' & \text{MK}(2) = 0 \end{array}$$

Cut set No. 2 is a component No. 1, but cut set No. 1 is gate 11. Therefore, we replace gate 11 in cut set 1 by gate 12 and 19.

$$\begin{array}{ll} \text{KM}(1) = \text{B}'\underline{12},\underline{19}' & \text{MK}(1) = 0 \end{array}$$

Gate 12 is an OR gate therefore, we increase the number of cut sets

$$\begin{array}{ll} \text{KM}(1) = \text{B}'2,\underline{19}' & \text{MK}(1) = 0 \\ \text{KM}(3) = \text{B}'3,\underline{19}' & \text{MK}(3) = 0 \\ \text{KM}(4) = \text{B}'\underline{13},\underline{19}' & \text{MK}(4) = 0 \end{array}$$

In cut set 1 we have gate 19 which is an OR gate, replacing it by its input we have:

$$\begin{array}{ll} \text{KM}(1) = B'2, \underline{20}' & \text{MK}(1) = 0 \\ \text{KM}(5) = B'2' & \text{MK}(5) = 0 \\ \text{KM}(6) = B'2, 4' & \text{MK}(6) = 0 \end{array}$$

We continue the replacement procedure until there is no gate in the cut set.

$$\begin{array}{ll} \text{KM}(1) = B'2, 6, \underline{14}' & \text{MK}(1) = 0 \\ \text{KM}(1) = B'2, 6, 15, \underline{16}' & \text{MK}(1) = 0 \\ \text{KM}(1) = B'2, 6, \underline{16}' & \text{MK}(1) = 0 \\ \text{KM}(7) = B'2, 6, \underline{16}, \underline{18}' & \text{MK}(6) = 0 \\ \text{KM}(1) = B'2, 6' & \text{MK}(1) = B'6' \\ \text{KM}(8) = B'2, 6, \underline{17}' & \text{MK}(8) = 0 \end{array}$$

At this stage, since the positive part of cut set number 1 is composed of basic events only, the code first combines the two parts of the cut set to nullify the cut sets which contain both complement and basic event of a component (i.e., $A.\bar{A} = 0$), see Appendix B.1. In this example, combination of KM and MK results in:

$$(\text{KM}, \bar{\text{MK}}) = B'2, 6, \bar{6}' = 0$$

The above process is executed in Subroutine MCSN1. Subroutine MCSN1 is also used to compare the new cut sets with the previously generated cut sets to check if it is an identical or super set. If it is neither, then the code calls

Subroutine MCSN2 to check for the existence of any super sets corresponding to new cut set. If so, it should be nullified. The basic logic used in these two subroutines is the combination of AND and OR logic, i.e., if CS1 and CS2 are the two cut sets to be compared and CS1 is the new one, then we have the following cases:

- a) IF (IAND(CS1,CS2).EQ.CS2) (Subroutine MCSN1)
CS1 is either identical or a super set to CS2
- b) IF(IAND(CS1,CS2).EQ.CS1) (Subroutine MCSN2)
CS2 is a super set of CS1

In case a), CS1 will be nullified whereas in case b), CS2 will be nullified.

After the comparison and elimination process, the code will start the substitution process by replacing the last cut set number for the one just eliminated. Here, cut set No. 8 will replace cut set No. 1 and the code will continue by substituting for gate 17.

$$KM(1) = B'2,6,5' \quad MK(1) = 0$$

The new cut set No. 8 is:

$$KM(8) = B'2,6,7' \quad MK(8) = 0$$

$$KM(9) = B'2,6,8' \quad MK(9) = 0$$

$$KM(10) = B'2,6,9' \quad MK(10) = 0$$

Again cut set No. 1 is a gate free cut set. Therefore, subroutine MCSN1 will be called. Since there exists cut set

No. 5 which has component No. 2, cut set No. 1, which is a super set of No. 5, will be replaced by cut set 10 and the process will continue. In fact, all the previously found cut sets which have component No. 2 in them will be cancelled. This elimination and substitution will continue until the code has swept completely through the fault tree and found all the minimal cut sets. The final results for this example are shown in Table 3.1.

The example in Table 3.1 was checked against the results of Bennett (1975) and Henley & Kumamoto (1978). It took BIT a fraction of second to generate all the cut sets (i.e., 0.07 sec.).

In another example, the code was checked against MODCUT. The fault tree used is shown in Appendix B.2. It generates the 86 cut sets in 0.84 seconds compared to MODCUT which takes 9.8 seconds.

In the third example, the code was checked against WAM-CUT. Here we do not have any processing time for WAM-CUT. The BIT code generates all the minimal cut sets in 0.35 seconds. The fault tree and the respective minimal cut sets are shown in Appendix B.2.

As we discussed earlier, cut set generation is a lengthy process and in order to shorten it one has to be familiar with the tree and be able to reduce it as much as possible. There are many methods that can be used to reduce the fault tree:

Table 3.1: MINIMAL CUT SET GENERATED BY BIT
FOR THE EXAMPLE FAULT TREE

<u>Cut Set No.</u>	<u>Components in the Cut Set</u>	
	<u>Name</u>	<u>No.</u>
1	A	1
2	B	2
3	CD	3,4
4	DEF	4,5,6
5	DFG	4,6,7,
6	D $\bar{F}$ H	4,6,8
7	D $\bar{F}$ I	4,6,9
8	C $\bar{E}$ F $\bar{G}$ H $\bar{I}$	3,5,6,7,8,9

- 1) Combine the successive gates as much as possible. However, this can only be done if the successive gates have the same type of gate.
- 2) Break down the large tree to small non-repetitive sub trees.
- 3) Combine some of the components inputted into a gate or different gates and represent them as a special component. This method is very useful in quantitative analysis. To determine the properties of a special component in terms of individual component properties, see Appendix B.3.

3.3 Unavailability Evaluator

3.3.1 Introduction

Quantitative evaluation of system reliability is concerned with synthesizing reliability characteristics for the TOP event from basic component reliability characteristics. Most published methods for quantitative evaluation require minimal cut sets as input to represent the system model.

Component failures represented in a logic model, i.e., fault tree or event tree, can either be statistically dependent or statistically independent. Conventional reliability analysis methods assume that component failures are statistically independent. However, dependencies can be easily modeled in the logic diagram so that the events in the final minimal cut sets are statistically independent, see Gokcek et al. (1979), Elerath & Ingram (1979), EPRI 217-2-2 (1975). One of the methods which can be used to model the dependencies is the

Markovian model. By this method, one can easily separate the dependent logic and show the combined system as a non-dependent event in the fault tree. But, one should bear in mind that the above model sometimes requires the use of a computer program for solving linear differential equations. In certain cases, the method discussed in Chapter 2 might be helpful in finding the system reliability function.

Another method for handling dependent events requires redrawing the fault tree. Suppose one wants to model the dependent event B in which the probability of event B is dependent on the occurrence of event A. Then one way to show the dependency is to use the building block shown in Fig. 3.6.

As it is shown in Appendix B.1, the event space in Fig. 3.6 represents the logical expression:

$$B = (A \cap B/A) \cup (\bar{A} \cap B/\bar{A})$$

Where:

B/A = Event B given the occurrence of A

And

$B/\bar{A}$ = Event B given the occurrence of $\bar{A}$

The above identity holds since, if we assume

$$C = (A \cap B/A) \cup (\bar{A} \cap B/\bar{A}) \quad \text{Then}$$

$$C = (A \cap B) \cup (\bar{A} \cap B) \quad \text{from the definition of events } B/A \text{ \& } B/\bar{A}$$

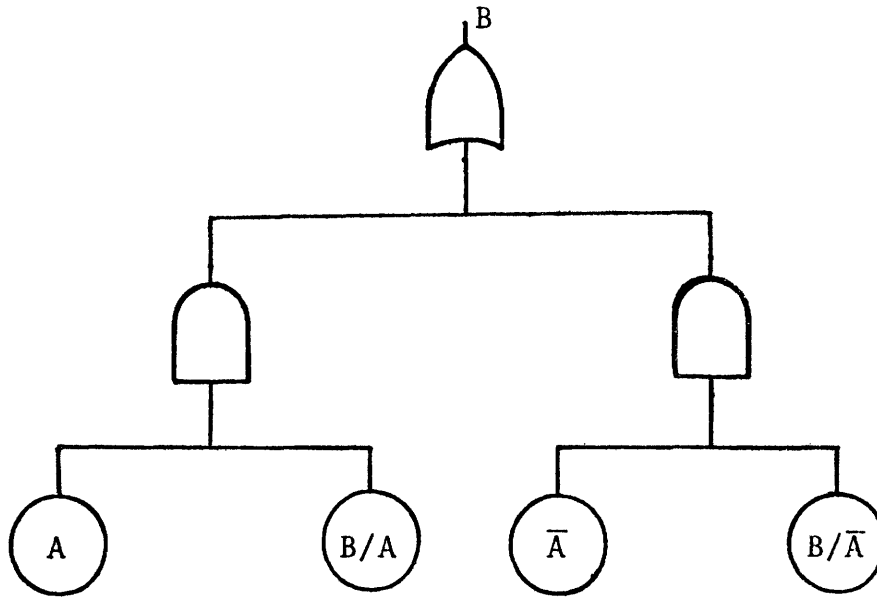


Fig. 3.6: Fault tree showing a dependence of B on the occurrence of A.

Thus:

$$C = B \cap (A \cup \bar{A}) = B$$

When applying this model to large fault trees, it must be recognized that the dependent event building block is irreducible.

Figure 3.6 could be extended to model common mode failure. For example, the interaction of a number of components to a common mode initiator (such as flooding or fire) can be described by incorporating a dependent event building block for each component. Figure 3.7 shows a model for describing the top event B of a system in which the common mode initiator event A is specified.

Knowing this, we can model our logic diagram i.e., fault tree, such that each basic event represents an independent event and we can develop certain mathematical models that will allow us to predict individual basic event unreliability and/or unavailability. Finally, by using the minimal cut set we can calculate the top event unreliability and/or unavailability.

The first comprehensive unreliability analysis of emergency safety features was performed in WASH 1400 (1975). In that study most of event occurrences were assumed to be constant. But, since most of emergency safety features are redundant systems and they are periodically tested to insure their readiness, one has to make a time dependent unreliability analysis to be able to predict the system unavailability more

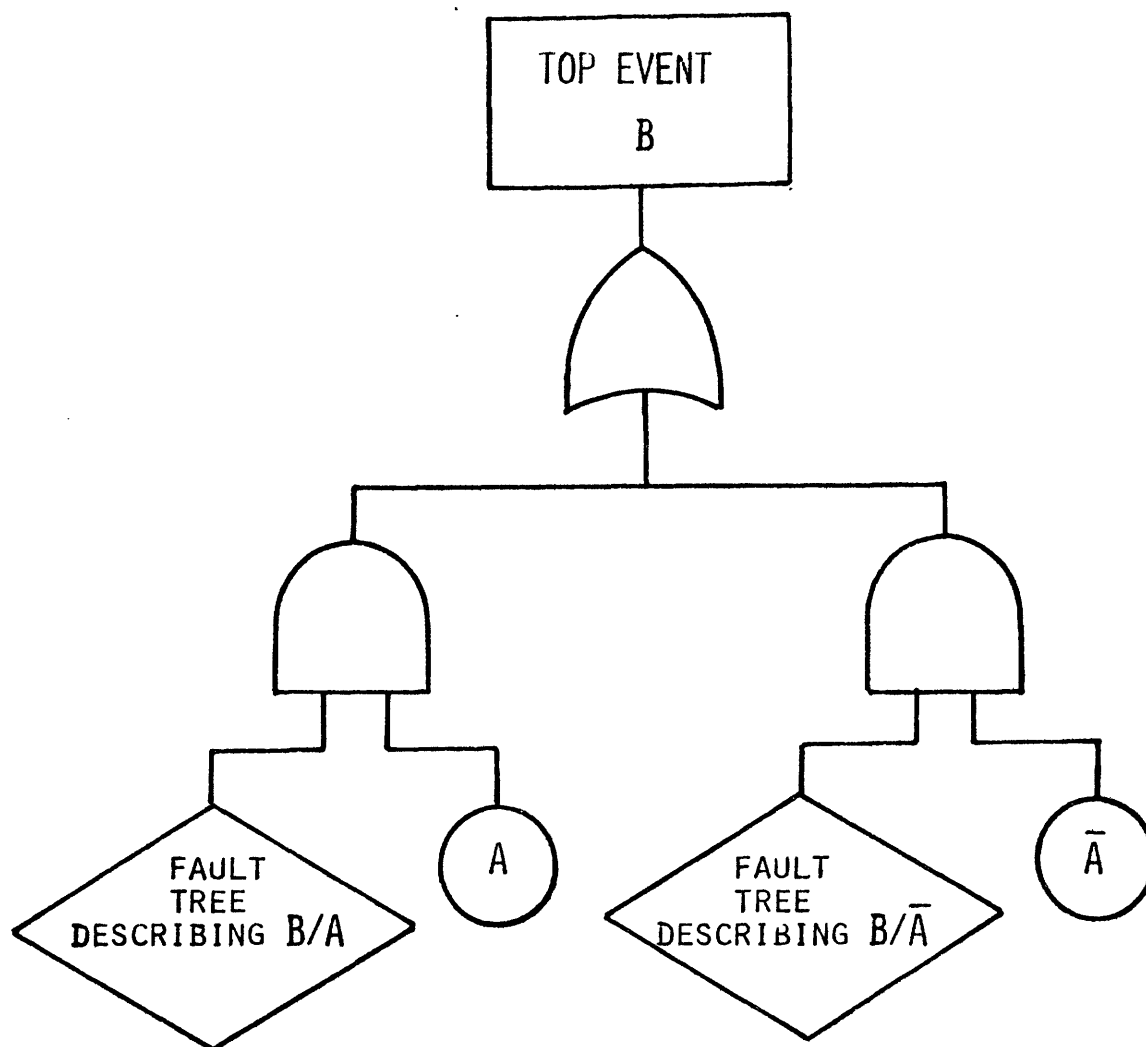


Fig. 3.7: Dependent event connecting fault tree.

realistically. In today's quantitative codes only KITT 1 & 2 (Vesely et al. (1970)), PL-MODT (Modarres (1979)) and FRANTIC (Vesely and Goldberg (1977)) are time dependent evaluators. KITT 1 uses exponential failure and repair distributions with constant parameters but is unable to analyze the periodically tested components. KITT 2 is the same as KITT 1 except that it has the capability of accommodating step wise time dependent repair and failure rates. FRANTIC is an approximated KITT code with the capability of handling periodically tested components. The above codes are written in FORTRAN IV. PL-MODT is a PL/1 code which uses modular techniques and has the same capability as the FRANTIC code. All the above codes have prespecified failure and repair distributions, and most of the time the distributions for the system being analyzed are not the same as those which have been coded. Also, some systems require more rigorous evaluation because of the existence of different types of failures and paths that the system may undergo during its change of state. For example, consider a pump. A pump can be in one of the following states at any given time:

- 1) up state (working or stand-by)
- 2) under the test
- 3) under repair

Because of certain mechanical difficulties or economical reasons not all the possible failure models can be immediately detected upon the failure, (i.e., it is impossible to allocate

a detector for crank shaft failure or it is uneconomical to install detectors for all the ancillaries). Therefore, there is a chance that the pump must go under the test in order to find its malfunction, or there is a chance with probability P_1 that the pump can be started by resetting its control panel. Figure 3.8 shows all the possible paths that the pump may undergo during its change of state.

3.3.2 Unavailability Evaluator Used in UNRAC

For our unavailability evaluator we used the models discussed in Chapter 2 for the maintained and non-maintained systems (components). For the maintained system, we consider a variety of distributions as a candidate for repair distribution and they are:

- 1) exponential
- 2) constant
- 3) Erlangian (special Gamma)
- 4) log normal (approximated by combination of exponentials)

The related reliability equation for a two-state system with exponential or constant distribution has been discussed earlier (cf. Eqns. 2.12 and 2.18).

In the case of periodically maintained components, the general equation developed in Chapter 2 (cf. Eqn. 2.38) has been used with the following repair distribution capabilities:

- 1) constant
- 2) exponential

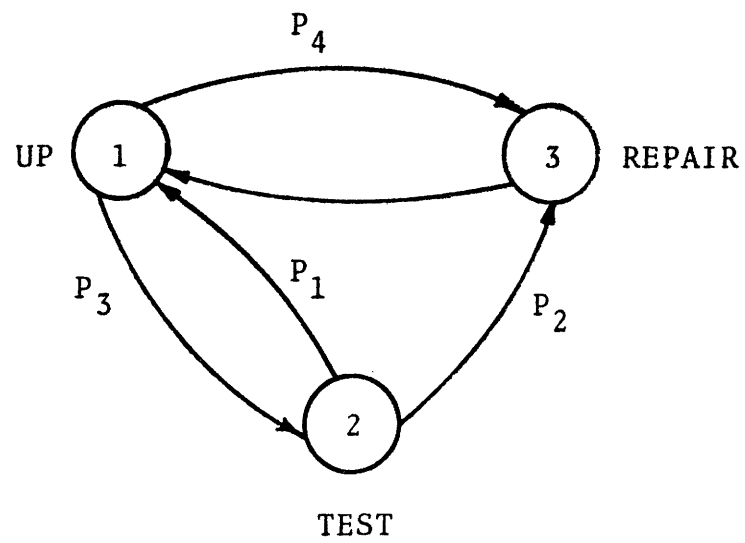


Fig. 3.8: All possible paths of a pump failure.

3) Erlangian

For the detailed equations see Appendix A.1.

To be able to analyze a 3 state component, see Fig. 3.8, we model the component as a semi-Markovian process. Figure 3.9 shows a detailed 3 state system with all the respective holding time distributions. Using the method discussed in Sections 2.3.1 and 2.3.2, we can show that the probability that the system will remain in state 1 at time t given that it was at state 1 at time zero have the following Laplace transforms:

$$\phi_{11}(s) = \frac{1 - F^*(s)}{s[1 - P_4 F^*(s) G^*(s) - P_3 F^*(s) P_1 - P_3 F^*(s) P_2 T^*(s) G^*(s)]} \quad (3.1)$$

The above equation can be easily transformed to equation (2.16) by putting $P_3=0$ and $P_4=1$. Therefore, we used this equation as a general equation for the monitored system (component). The application of equation (3.1) to the aforementioned repair distributions are:

- 1) If we assume repair and test distribution to be exponential then we have

$$\begin{aligned} G^*(s) &= \frac{\mu}{\mu+s} \\ T^*(s) &= \frac{\nu}{\nu+s} \\ F^*(s) &= \frac{\lambda}{\lambda+s} \end{aligned} \quad (3.2)$$

where λ , μ , ν , are failure, repair, test completion rates and they are assumed to be constant. Hence, by

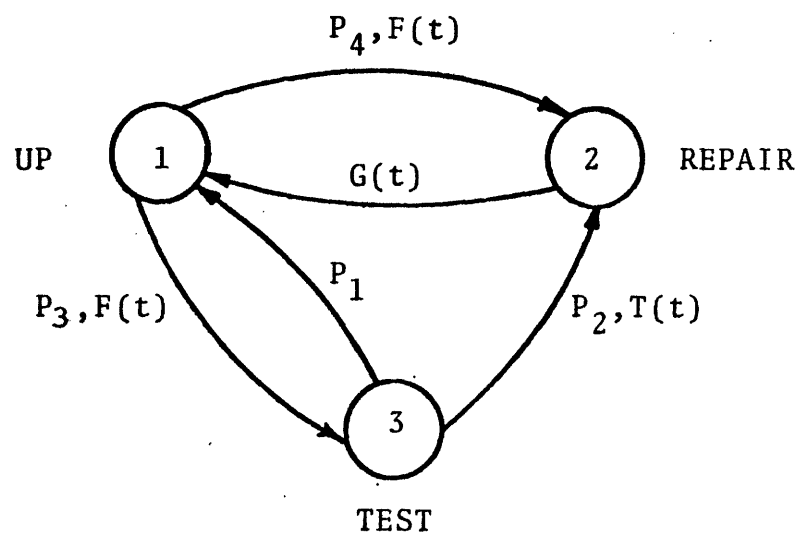


Fig. 3.9: A general three-state component.

substituting for $G^*(s)$, $F^*(s)$ and $T^*(s)$ in Eqn. (3.1) we get:

$$\phi_{11}(s) = \frac{(\mu + s)(\nu + s)}{s[s^2 + as + b]} \quad (3.3)$$

where

$$a = \lambda + \nu + \mu - P_1 P_3 \lambda$$

$$b = \lambda \nu + \mu \nu + P_3 \lambda \mu - P_1 P_3 \lambda (\nu + \mu)$$

Inversion of Eqn. (3.3) to the t domain results in a series of equations which are given in Appendix A.2. The asymptotic value of $\phi_{11}(t)$ can be easily shown by applying final value theorem:

$$\phi_{11}(\infty) = \lim_{t \rightarrow \infty} \phi_{11}(t) = \lim_{s \rightarrow 0} [s \phi_{11}(s)] \quad (3.4)$$

Hence

$$\phi_{11}(\infty) = \frac{\mu \cdot \nu}{b} \quad (3.5)$$

- 2) If we assume that the repair distribution is a 2nd order Erlangian then $\phi_{11}(s)$ will be:

$$\phi_{11}(s) = \frac{(\mu + s)^2 (\nu + s)}{s[s^3 + as^2 + bs + c]} \quad (3.6)$$

where

$$a = \lambda + \nu + 2\mu - P_1 P_3 \lambda$$

$$b = 2(\lambda + \nu)\mu + \mu^2 + \nu\lambda - P_1 P_3 (2\mu + \nu)\lambda$$

$$c = (\nu + \lambda)\mu^2 + 2\mu\nu\lambda - P_3 P_1 \lambda \mu (2\nu + \mu) - P_4 \lambda \mu^2$$

Again we only show the asymptotic value here and the rest of the respective equations are given in Appendix A.2.

$$\phi_{11}(\infty) = \frac{\mu^2 \cdot \nu}{c} \quad (3.7)$$

- 3) If we assume the repair distribution to be lognormal, we can approximately match it with a combination of exponential distributions, i.e. we can assume the repair distribution, $g(t)$, to have the following form:

$$g(t) = A[e^{-\lambda_1 t} - e^{-\lambda_2 t}] \quad (3.8)$$

where

$$A \text{ by normalization is equal to } \frac{\lambda_2 \lambda_1}{\lambda_2 - \lambda_1}$$

$$\text{and } G^*(s) = \frac{\lambda_1 \lambda_2}{(\lambda_1 + s)(\lambda_2 + s)} \quad (3.9)$$

using the above equation for $G^*(s)$, the asymptotic value for $\phi_{11}(t)$ will be:

$$\phi_{11}(\infty) = \frac{\lambda_1 \lambda_2 \nu}{c} \quad (3.10)$$

where

$$c = (v + \lambda)\lambda_1\lambda_2 + (\lambda_1 + \lambda_2)v\lambda - P_1P_3(\lambda_1v + \lambda_2v + \lambda_1\lambda_2)\lambda \\ - P_4\lambda_1\lambda_2\lambda$$

and the rest of the related equations are given in Appendix A.2.

Since the cumulative repair probability distributions signifies the completeness of the repair, we compare the lognormal and its approximated (matched) combination of exponentials by their cumulative distribution values. Figures 3.10 to 3.12 show the above comparison for two typical examples. As can be seen, for a highly skewed lognormal (large σ), the difference between the results is on the order of maximum 2% (Figs. 3.10, 3.12). But for a small standard deviation (small σ), the lognormal repair distribution approaches 1 faster than its approximated distribution. Therefore, the combination of exponentials produces reasonably accurate (for large σ) and/or conservative (for small σ) results. For detailed values of Fig. 3.10 to 3.12 and for the method for approximating lognormal, see Appendix A.2.

Having prepared the reliability equations for all the possible types of components involved in a system, we need a general system function to be able to evaluate the systems reliability. As we mentioned earlier, most of the quantitative codes use minimal cut sets as input to represent the system model and the general system function given the minimal cut

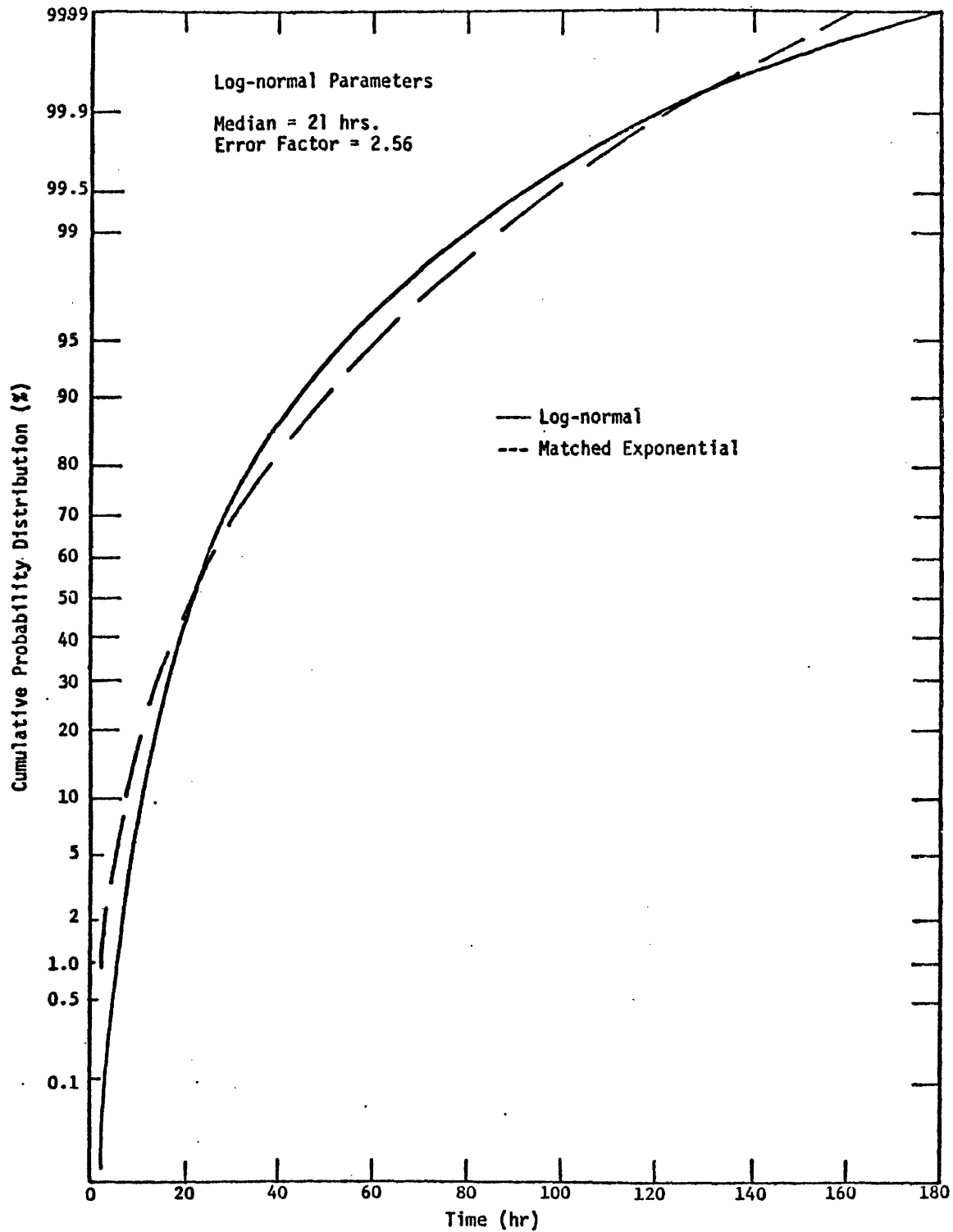


Fig. 3.10: Comparison of cumulative probability distribution of a log-normal with its approximated combination of exponentials (median of 21 (hrs) and error factor of 2.56)

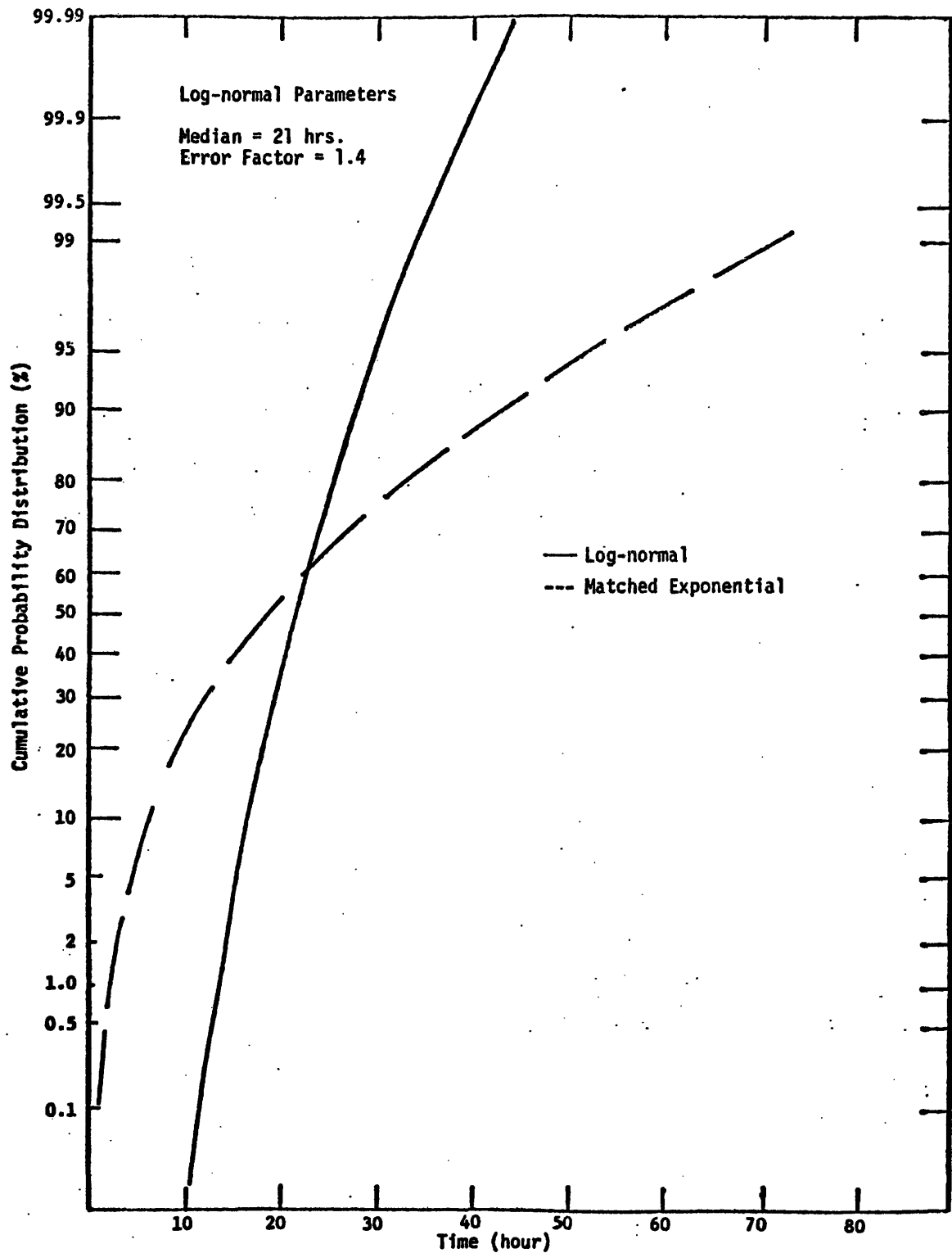


Fig. 3.11: Comparison of cumulative probability distribution of a log-normal with its approximated combination of exponentials (median of 21 (hrs) and error factor of 1.4)

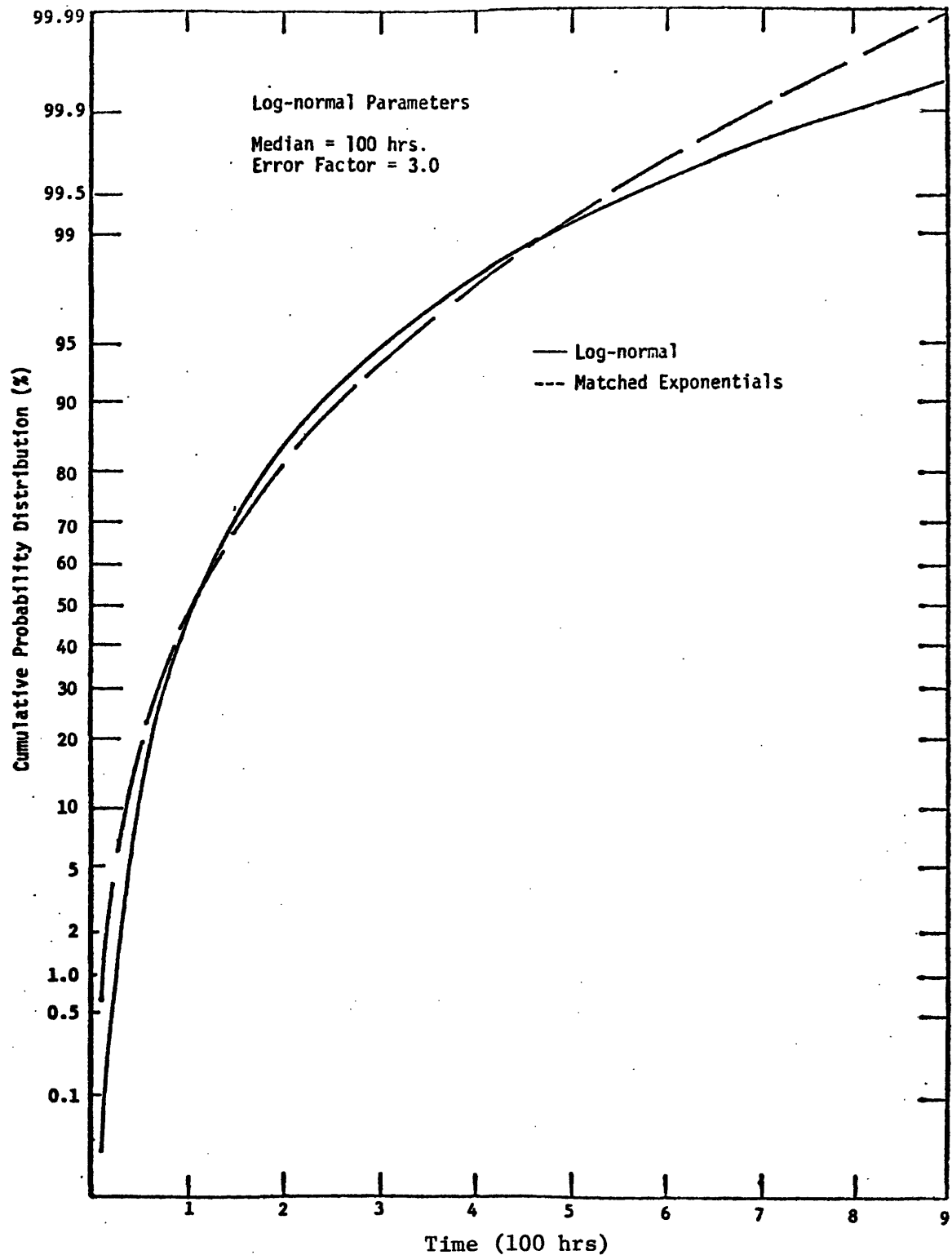


Fig. 3.12: Comparison of cumulative probability distribution of a log-normal with its approximated combination of exponentials (median of 100 (hrs) and error factor of 3)

sets has the following form:

$$Q_{\text{Top}} = \bigcup_{s=1}^N \bigcap_{j \in s} q_j \quad (3.11)$$

where:

q_j is the component unavailability or unreliability

j is the component number in s cut set

N is the total number of cut sets

$\bigcup$ is the union and by definition:

$$\bigcup_{i=1}^m Y_i = 1 - \prod_{i=1}^m (1 - Y_i); \quad Y_i \text{ is any function}$$

Expanding Eqn. (3.11) in terms of cut set unavailability Q_{c_i} , ($Q_{c_i} = \bigcap_{j \in s} q_j$), we get:

$$Q_{\text{Top}} = \sum_{i=1}^N Q_{c_i} - \sum_{\substack{i=1 \\ i < j}}^N Q_{c_i} Q_{c_j} + \sum_{\substack{i=1 \\ i < j < l}}^N Q_{c_i} Q_{c_j} Q_{c_l} - \dots$$

$$+ (-1)^N \prod_{i=1}^N Q_{c_i} \quad (3.12)$$

Evaluating the Q_{Top} for a medium size fault tree requires a lot of computation time. Therefore, we approximate Eqn.(3.12) by the method of inclusion-exclusion principle discussed by Feller (1957) and Vesely (1970).

- a) first term approximation, sometimes called rare-event approximation

$$Q_{\text{Top}} \approx \sum_{i=1}^N Q_{c_i} = s_1 \quad (3.13)$$

b) first three terms approximation

$$Q_{\text{Top}} \approx s_1 - s_2 + s_3 \quad (3.14)$$

c) first 5 terms approximation

$$Q_{\text{Top}} \approx s_1 - s_2 + s_3 - s_4 + s_5 \quad (3.15)$$

where

s_i is the i th term in Eqn. (3.12).

The degree of approximation required to find the Q_{Top} is usually dependent on the component unreliability and overall size of cut sets involved. For low minimal cut set size, we sometimes have to use a higher degree of approximation to get a more accurate result. Occasionally, the following equation is used to find the range of the top event unavailability.

$$s_1 - s_2 < Q_{\text{Top}} < s_1 \quad (3.16)$$

To prepare our unavailability evaluator, we restructured FRANTIC (Vesley and Goldberg (1977)) to incorporate our models. We call the package UNReliability Analysis Code or UNRAC. The code is organized in such a way that any user with a limited knowledge of the computer can follow the code should an alteration or modification be deemed necessary. Figure 3.13 shows the logical flow chart used in UNRAC. For more information about the structure of the code see Appendix C.

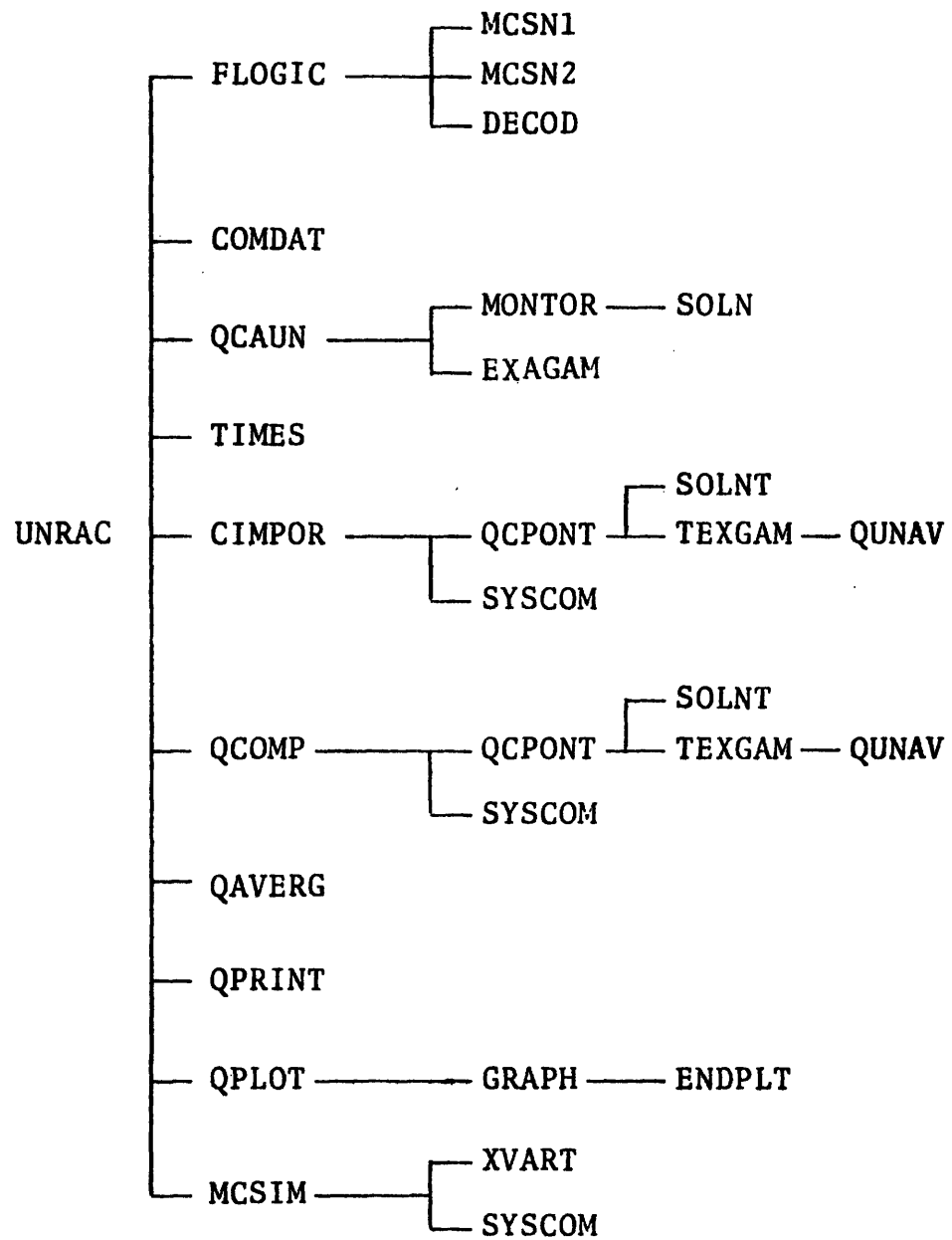


Fig. 3.13: Logical flow chart used in UNRAC.

In order to check the accuracy of the code we benchmarked it against PREP and KITT and PL-MODT. Figure 3.14 shows a sample fault tree and respective failure rates which appeared in Vesely (1970) and Modarres (1979). Figure 3.15 shows the results of this study with PREP and KITT, PL-MODT and BIT-FRANTIC (Wolf and Karimi (1978)). As we can see, the results of UNRAC are more consistent with the exact results than either FRANTIC or PL-MODT. In fact, with the use of the first 3 terms approximation, UNRAC produces almost identical results to the ones given in Vesely (1970), see Table 3.2.

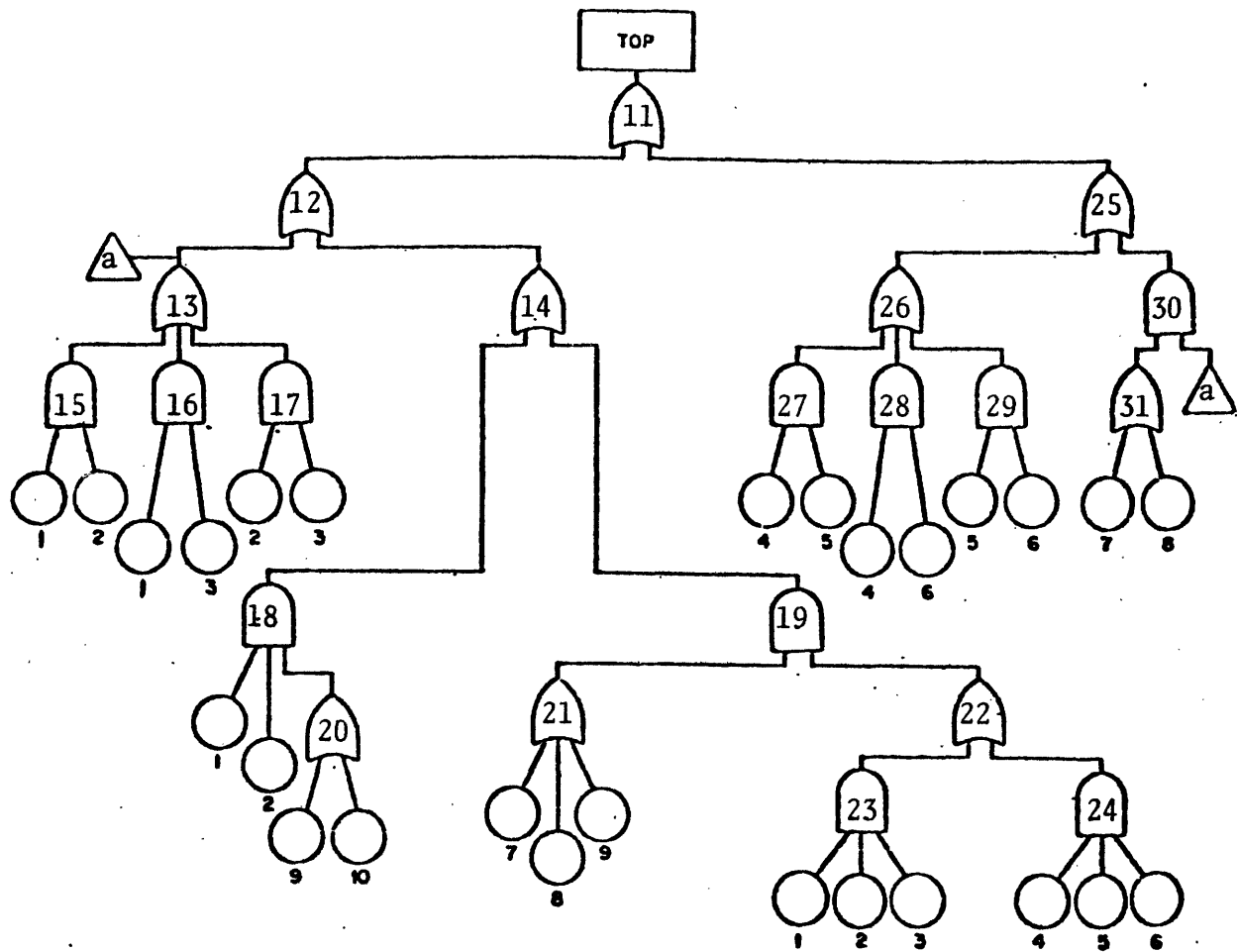
3.4 Uncertainty Bounds Evaluator

3.4.1 Introduction

The unreliability of the top event of a fault tree is evaluated from the unreliabilities of the basic events contained in the tree. This quantification is a straight forward process when the basic events (components) parameters (failure rate, repair rate, etc.) are exactly known. However, uncertainties exist in components' data because:

- 1) lack of consistent information on the failure and repair rates,
- 2) diversity in manufacturing specifications.

Therefore, any quantification analysis is incomplete without information on the uncertainty of the top event unreliability. There are two general methods of propagating uncertainties. One of these is an analytic approach, Lee and Apostalakis (1976), where the authors first use the Johnson family



Min. Cut Sets

1,2
1,3
2,3
4,5
4,6
5,6

Component Failure Rate

<u>Comp. #</u>	<u>$\lambda \times 10^6 \text{hr}^{-1}$</u>
1-3	2.6
4-6	3.5
7,8	5.0
9,10	8.0

Fig. 3.14: Fault tree diagram. Minimal cut sets and failure data of an example Reactor Protection System (Vesely (1970)).

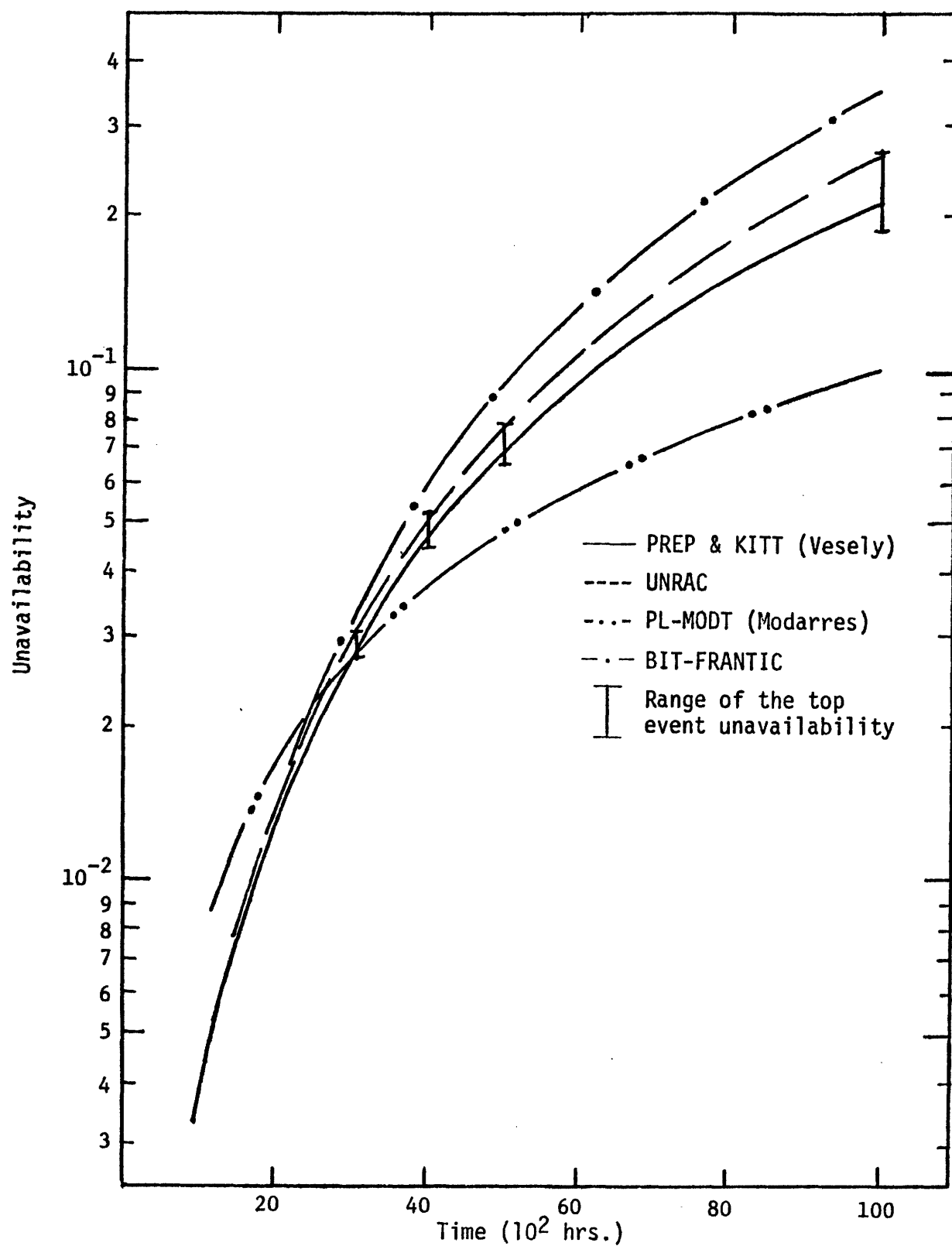


Fig. 3.15: A comparison of UNRAC with PREP & KITT, PL-MODT, and BIT-FRANTIC.

Table 3.2 COMPARISON OF THE RESULTS OF THIS STUDY WITH
RESULTS OF KITT FOR AN EXAMPLE FAULT TREE

Time $\times 10^3 \text{ hr}$	Vesely (1970)	1st Term Approx.	1st 3 terms Approx.
	Exact Q	Q	Q
0.0	0.0	0.0	0.0
1.0	3.49 - 03	3.569 - 03	3.488 - 03
2.0	1.32 - 02	1.379 - 02	1.317 - 02
3.0	2.80 - 02	2.998 - 02	2.80 - 02
4.0	4.70 - 02	5.152 - 02	4.704 - 02
5.0	6.95 - 02	7.782 - 02	6.95 - 02
10.0	2.12 - 01	2.636 - 01	2.117 - 01

distributions, (see Hahn and Shapiro (1967)), to approximate each components' distributions, then find the first two moments of the top event function. Finally they use these two moments to estimate confidence bounds by standard inequalities (e.g. Tchebychev, Cantelli) or empirical distributions (moment matching technique). This method is only useful for small fault trees.

The other approach is the Monte-Carlo simulation routine which is presented in this section and can be used to obtain an empirical Bayes estimation of the probability interval of the system reliability (unreliability). In this procedure, each components failure probability is represented by an appropriate distribution. And in order to evaluate the top event probability, each components failure probabilities will be simulated randomly according to their respective distributions. One of the drawbacks of this technique is that a high degree of resolution for the top event distribution, requires that the sampling be done a large number of times.

Error propagation by means of Monte-Carlo simulation has previously been investigated in the Reactor Safety Study, [WASH 1400 (1975)] and many ideas and assumptions used in this study are adopted from that report. Several codes have been written to compute the probability distribution of the top event, i.e. SAMPLE (WASH 1400 (1975)), LIMITS (Lee and Salem (1977)), MOCARS (Matthews (1977)).

All the above mentioned codes require user-supplied

function as input to identify the logical dependence of the components in the system that is being analyzed. For a very large fault tree and/or for a medium tree but with a lot of repetitive events, the construction of this equation is a non-trivial process and there is a high probability of introducing errors. Therefore, there is an incentive to have a Monte-Carlo simulator coupled with our unreliability analysis code.

3.4.2. Monte-Carlo Simulator Used in UNRAC

The Monte-Carlo routine used in conjunction with UNRAC is called MCSIM (Monte-Carlo SIMulation). Here we will cite the mathematical concept of the Monte-Carlo simulation without proof, and the reader will be referred to Dixon and Massey (1957), Cashwell and Everett (1959), Hammersley and Handscomb (1964), and Mann et al. (1974) for details.

MCSIM allows simulation to be carried out for fault trees having the following component types:

- 1) components with constant failure probability
- 2) non-maintained component
- 3) monitored component
- 4) periodically tested components.

And it can accomodate the following types of distributions for the component's parameters:

- 1) exponential
- 2) lognormal
- 3) gamma
- 4) normal

It also has the following capabilities:

- 1) permits simulation for both the average and the time dependent unreliability, and
- 2) allows propagation of error bounds on the following failure parameters:
 - a) failure rate
 - b) mean repair time
 - c) mean test time.

All the aforementioned properties are encoded in the XVART routine.

In general, there exist two steps in every Monte-Carlo simulation code used in reliability analysis. First, one simulates the desired distribution and then sorts the results in an ascending or descending order. Therefore, there exist two time consuming steps that one has to overcome in order to have a comparatively efficient code. For the first step, numerous reports on accelerating random number generation have been written. Among these are: Conveyou and MacPherson (1967), MacLaren, et al. (1964), Marsaglia, et al. (1964a) and (1964b). The above authors discuss the fast methods to generate Uniform, Exponential and Normal random numbers. And, in general, these are sufficient to generate any kind of distributions random numbers. For example, if one wants to generate a Gamma random number, one needs a combination of Exponential and Uniform random numbers. In the case of Normal random numbers, one can use either the Central Limit Theorem

or use a combination of Uniform random numbers. Marsaglia et al. (1964a) suggest that the latter method needs less storage than the former one. Here we discuss how to generate a Uniform random number between zero and one, $U(0,1)$.

According to McGrath et al. (1975) for computers with an available word length less than or equal to 32 bits, the most efficient random number generator has the following form:

$$X_{n+1} = a X_n \pmod{2^P} \quad (3.17)$$

where

P is the number of bits (excluding sign) in a word;

for IBM 370, $P = 31$

a is called the generator which should be carefully selected

$X_i \pmod{2^P}$ is the remainder of $X_i/2^P$

The above routine for generating a random number is fast and will produce numbers whose properties approximate randomness sufficiently close for valid use in Monte-Carlo simulation; provided that special care is taken in choosing a proper generator. McGrath et al. (1975) suggest $a = 5^{13}$. In general, X_i repeats itself after at most 2^P steps. For further information about different distributions random number generators, see Appendix C.

In order to accelerate the sorting routine, we use a procedure developed by Shell (1956) and used in LIMITS Code. Figure 3.16 shows the difference between SAMPLE and LIMITS

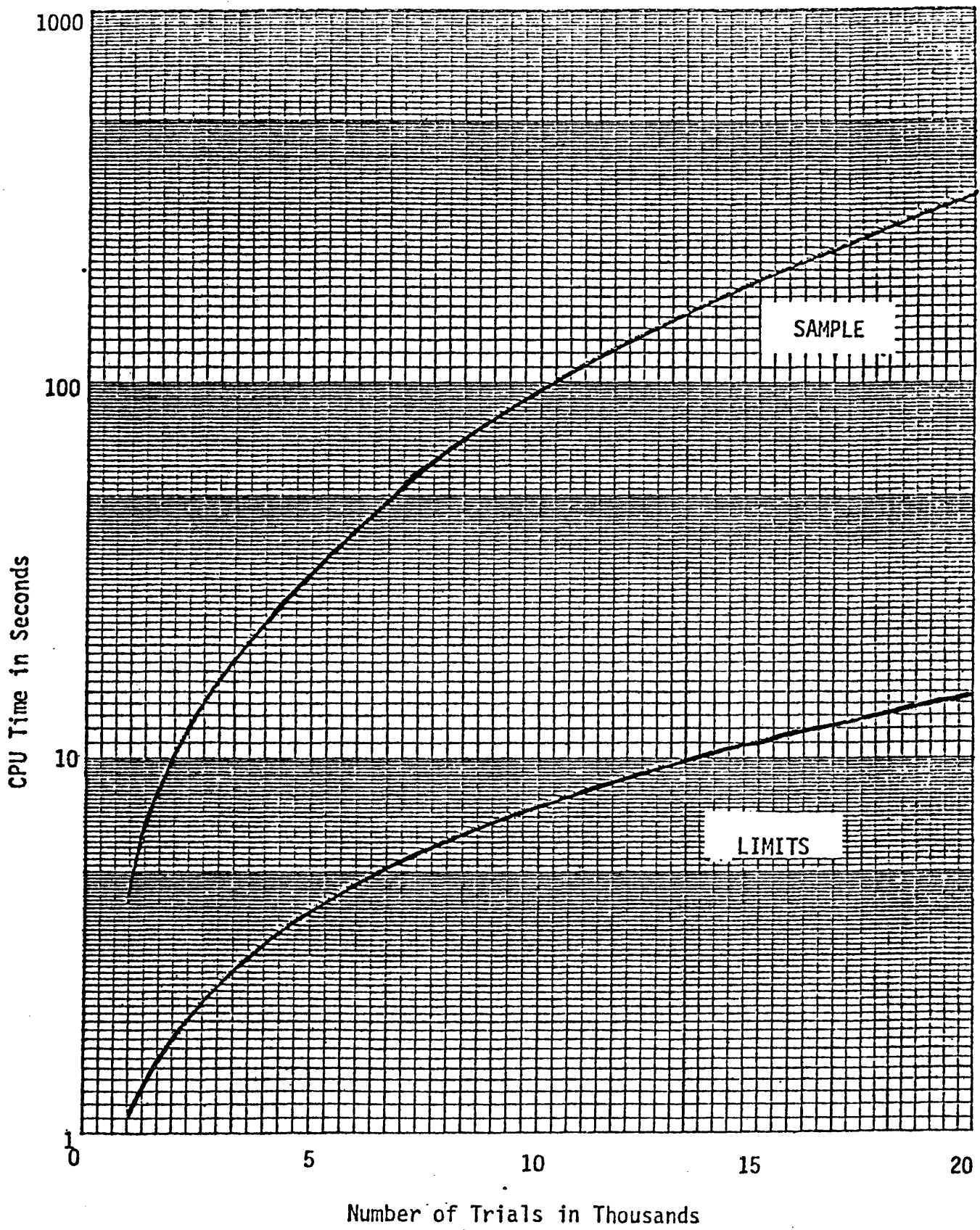


Fig. 3.16: The execution time of the LIMITS and the SAMPLE Codes (Lee & Salem (1977)).

execution time for a special case which appeared in Lee and Salem (1977). The flow chart of the sorting routine is shown in Appendix C.

Another important consideration of the Monte-Carlo simulation is the accuracy. The accuracy of a simulation can be indicated in several ways. An example of these is the simple non-parametric measurement of the accuracy of an estimated distribution by sampling given by Dixon and Massey (1957). An example of their accuracy measure for large sample size N can be expressed as:

$$P_r(|X_s - X_p| \leq \frac{1.36}{\sqrt{N}}) = 0.95 \quad (3.18)$$

Where X_s is the estimated distribution fractile, and X_p is the corresponding exact cumulative distribution value of the underlying population from which the sample was randomly picked. This measure of accuracy is too conservative and requires a large number of samples. For example, if we want to be 95% sure that the estimated distribution does not deviate more than 0.01 from the exact distribution throughout the entire range, a sample size of not less than 18,496 is required. Cashwell and Everett (1959) found the following equation by using Tchebychev's inequality and binomial sampling:

$$P_r(|\frac{M}{N} - P| \leq \epsilon) = \sqrt{\frac{2}{n}} \int_0^t e^{-\frac{u^2}{2}} du + R \quad (3.19)$$

Where N is the number of trials M is the number of success,

$t = \leq \frac{\sqrt{N}}{pq}$ and $q = 1-P$. The quantity R is the error associated with the probability measure and is given by:

$$|R| \leq \frac{e^{-\frac{t^2}{2}}}{\sqrt{2\pi Npq}} + \frac{0.2 + 0.25 |p-q|}{Npq} + e^{-(\frac{3}{2})\sqrt{Npq}} \quad (3.20)$$

Equation 3.19 can be put into a more familiar format:

$$P_r(|\frac{M}{N} - P| \leq \epsilon) = \text{erf}(\frac{t}{\sqrt{2}}) + R \quad (3.21)$$

The above set of equations do not have any specific limitation as to where it can be used. They are applicable to any distribution as long as the sampling is taken from an identically distributed random variable.

In Eqn. 3.21, $\frac{M}{N}$ is the actual probability limit associated with X_p which is the P fractile point in our simulated distribution, and if we denote it as $X(P)$, then we can write:

$$P_r(|X(P) - P| \leq \epsilon) = \text{erf}(\frac{t}{\sqrt{2}}) + R \quad (3.22)$$

For a given number of trials N , and P , the above probability is constant and will not vary with distribution. For example, if we fix $N = 2000$, $P = 0.95$ and $\epsilon = 0.4\%$, then

$$P_r(|X(0.95) - 0.95| \leq 0.004) = \text{erf}(0.58) = 0.5879$$

$$|R| = 0.039$$

which indicates that with probability 58.79%, the value calculated for the 95% confidence limit lies between 94.6% and 95.4% confidence limits. The maximum error in this case is 3.9%.

MCSIM gives this information with the final top event unavailability.

3.5 On the Distribution of Input Variables

This section is devoted to a description of the various forms of distributions that are used in this work for expressing the unreliabilities of input variables, i.e. failure rates, repair rates, etc. The question of how these forms can be obtained from existing information, theoretical models, engineering judgement or subjective considerations, is not addressed in this study and the reader will be referred to Mann et al. (1974).

The input variables of the various components of a system are assumed to be positive random variables taking values in any interval of the positive real axis. The following are the list of Pdf considered in various parts of this study.

A. Gamma Probability Density Function (pdf)

The gamma pdf is used to describe the distribution of continuous random variable bounds at one end. The gamma Pdf is defined by:

$$f_Y(t; \eta, \lambda) = \begin{cases} \frac{\lambda^\eta}{\Gamma(\eta)} t^{\eta-1} e^{-\lambda t} & t \geq 0, \lambda > 0, \eta > 0 \\ 0 & \text{elsewhere} \end{cases} \quad (3.23)$$

where $\Gamma(\eta)$ is the complete gamma function defined by

$$\Gamma(\eta) = \int_0^\infty x^{\eta-1} e^{-x} dx \quad (3.24)$$

and if η is a positive integer then:

$$\Gamma(\eta) = (\eta - 1)! \quad (3.25)$$

In the definition of the gamma distribution Eqn. (3.23) η can take on any positive value. When η is restricted to integers, the gamma distribution has been referred to as Erlangian Distribution. When $\eta = 1$, gamma distribution will be an Exponential distribution. In this study we used a general second order Erlangian distribution and an exponential distribution whose pdf's are defined by:

a) 2nd order Erlangian, $\eta = 2$

$$f(t; \lambda) = \begin{cases} \lambda^2 t e^{-\lambda t} \\ 0 \end{cases} \quad \text{elsewhere} \quad t \geq 0, \lambda > 0 \quad (3.26)$$

b) exponential

$$f(t; \lambda) = \begin{cases} \lambda e^{-\lambda t} \\ 0 \end{cases} \quad \lambda > 0, t \geq 0 \quad (3.27)$$

B. The Lognormal Probability Density Function

The lognormal pdf describes the distribution of a continuous random variable bounded from below by zero and whose logarithm is distributed according to a normal pdf with parameter μ and σ . The normal distribution has the following Pdf:

$$f_N(t; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2\sigma^2} (t - \mu)^2\right]$$

$$\sigma > 0 + \infty < \mu < -\infty \quad (3.28)$$

Thus, the lognormal pdf is defined by:

$$f_1(t; \mu, \sigma) = \begin{cases} \frac{1}{\sigma t^2} \exp\left[-\frac{1}{2\sigma^2} (\ln t - \mu)^2\right] & t > 0, \sigma > 0, \\ 0 & \text{elsewhere} \end{cases} \quad (3.29)$$

$$-\infty < \mu < \infty$$

The first four moments of the above distributions are given in Table 3.3.

To generalize the equations (3.24, 2.36, 2.37, 2.28, and 3.29) in order to describe a random variable that has an interval other than $[0,1]$, say $[X,1]$, we replace t by $(t-X)$ in the aforementioned equations.

3.6 Moment Matching Method

A distribution is completely defined once all its moments are known. However, many distributions can be adequately described by the first four moments. In addition, the first four moments play an important role in fitting empirical distribution and in approximating the distribution of a random variable.

Wilks (1962) argues that probability density functions (pdf) of bounded random variables with a finite number of lower moments in common exhibit similarities, since in the limit (all moments the same) they would coincide in a unique pdf. The unreliability of a system being a probability, is bounded because it can take values only in the interval $[0,1]$, and therefore, the moment matching can be applied if its n first moments are known. Obviously, the more moments that are

Table 3.3: SUMMARY OF IMPORTANT CHARACTERISTICS OF A SELECTED DISTRIBUTION

Dist. Type	Expected Value μ_1	Variance μ_2	$\sqrt{\beta_1} = \frac{\mu_3}{(\mu_2)^{3/2}}$	$\beta_2 = \frac{\mu_4}{\mu_2^2}$
Gamma	n/λ	n/λ^2	$2/\sqrt{n}$	$\frac{3(n+2)}{n}$
2nd order erlangian	$2/\lambda$	$2/\lambda^2$	$\sqrt{2}$	6
Exponential	$1/\lambda$	$1/\lambda^2$	2	9
Normal	μ	σ^2	0	3
Log-Normal	$e^{\mu + \frac{1}{2}\sigma^2}$	$e^{2\mu + \sigma^2} [e^{\sigma^2} - 1]$	$[e^{\sigma^2} - 1]^{\frac{1}{2}} (e^{\sigma^2} + 2)$	$3 + (X-1)(X^3 + 3X^2 + 6X + 6)$ where $X = e^{\sigma^2}$

available, the more exact the approximation would be. In most instances, however, the first four moments are adequate. This is the case when a two parameters pdf (like the ones presented in Section 3.4.1 or a member of the Johnson or Pearson families) is chosen as an approximation. The third and fourth moments determine the shape or the "type" of the distribution and the first two define its parameters. More precisely, the shape of the distribution is partly characterized by: (1) its third central moment or Skewness which is a measure of the asymmetry of the distribution, and (2) its fourth central moment or Kurtosis which is related to its peakedness. In order to make these two "measures" of the shape of a pdf independent from its scale, the following coefficients are defined:

$$\text{coefficient of Skewness: } \sqrt{\beta_1} \equiv \frac{\mu_3}{(\mu_2)^{3/2}} \quad (3.30)$$

and

$$\text{coefficient of Kurtosis: } \beta_2 \equiv \frac{\mu_4}{\mu_2^2} \quad (3.31)$$

When μ_k denotes the kth central moment of a random variable or:

$$\mu_k = \int_0^1 [\bar{R} - \mu_1']^k f(\bar{R}) d\bar{R} \quad (3.32)$$

Where

$f(\bar{R})$ is the pdf of $\bar{R}$ (unreliability or unavailability)

μ_1' denotes the first moment about the origin and is:

$$\mu_1' = \int_0^1 \bar{R} f(\bar{R}) d\bar{R} \quad (3.33)$$

If we denote μ_k' to be kth moment about the origin or

$$\mu_k' = \int_0^1 \bar{R}^k f(\bar{R}) d\bar{R} \quad (3.34)$$

Then we have:

$$\mu_2 = \mu_2' - (\mu_1')^2 \quad (3.35)$$

$$\mu_3 = \mu_3' - 3\mu_2' (\mu_1' + 2(\mu_1')^3) \quad (3.36)$$

$$\mu_4 = \mu_4' - 4\mu_3' + 6\mu_2'(\mu_1')^2 - 3(\mu_1')^4 \quad (3.37)$$

In cases where the form and parameters of the pdf under study is unknown, the central moments, μ_k , may be calculated by replacing μ_k' by $(\frac{1}{n}) \sum_{i=1}^n X_i^k$ in the preceding expressions, where X_i , $i=1,2,\dots,n$, are the values of n given observations (i.e., simulated top event reliability in Section 3.4). Thus, μ_2 will read:

$$\mu_2 = \frac{1}{n} \sum_{i=1}^n X_i^2 - \frac{1}{n^2} \left(\sum_{i=1}^n X_i \right)^2 \quad (3.38)$$

The above equation leads to what statisticians call a "biased estimate." Therefore, we use the corresponding unbiased formula which is:

$$\mu_2 = \frac{n \sum_{i=1}^n X_i^2 - (\sum_{i=1}^n X_i)^2}{n(n-1)} \quad (3.39)$$

to estimate the variance of the distribution.

Hence, if the coefficient β_1 and β_2 can be obtained, the shape of the distribution is approximately defined. Fig. 3.17 and Fig. 3.18 give numerical values of the coefficient and

of the various "theoretical" type of densities presented in Section 3.4.1, and Johnson distributions (see Hahn and Shapiro (1967)) respectively. From these figures, the type of density that has the same β_1 and β_2 with the sought pdf can be obtained.

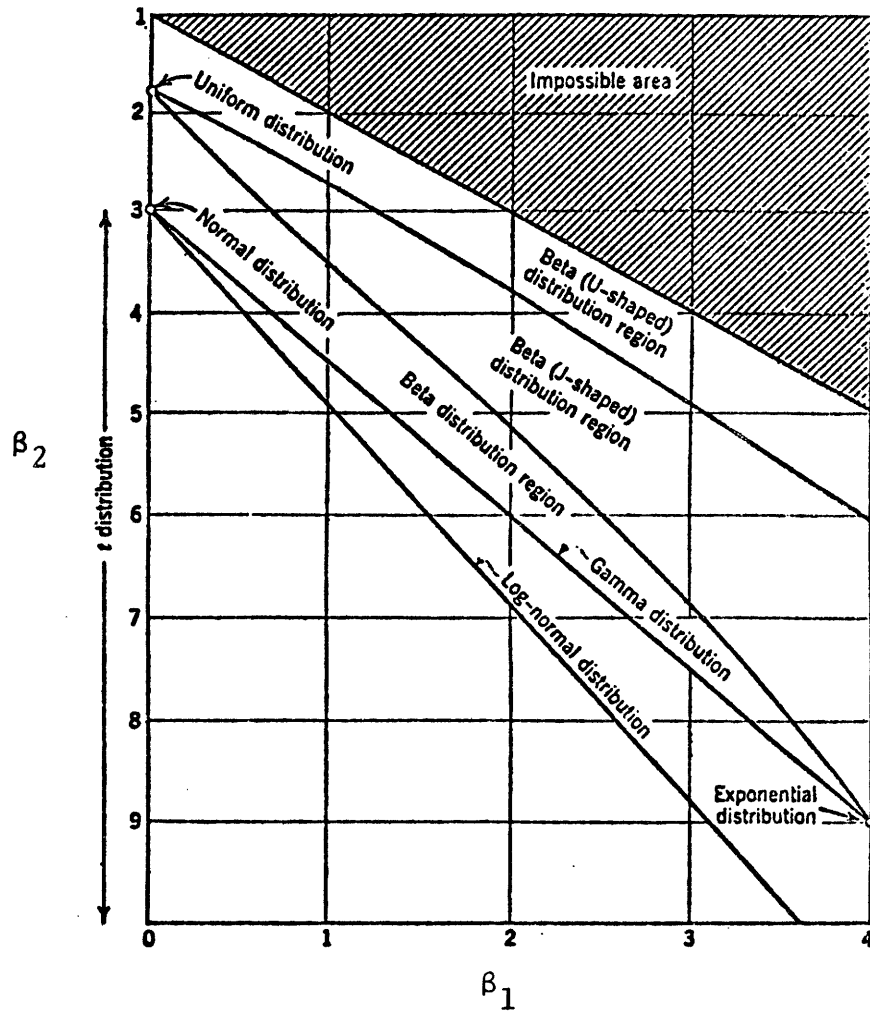


Fig. 3.17: Region in (β_1, β_2) plane for various distributions (Hahn & Shapiro (1967)).

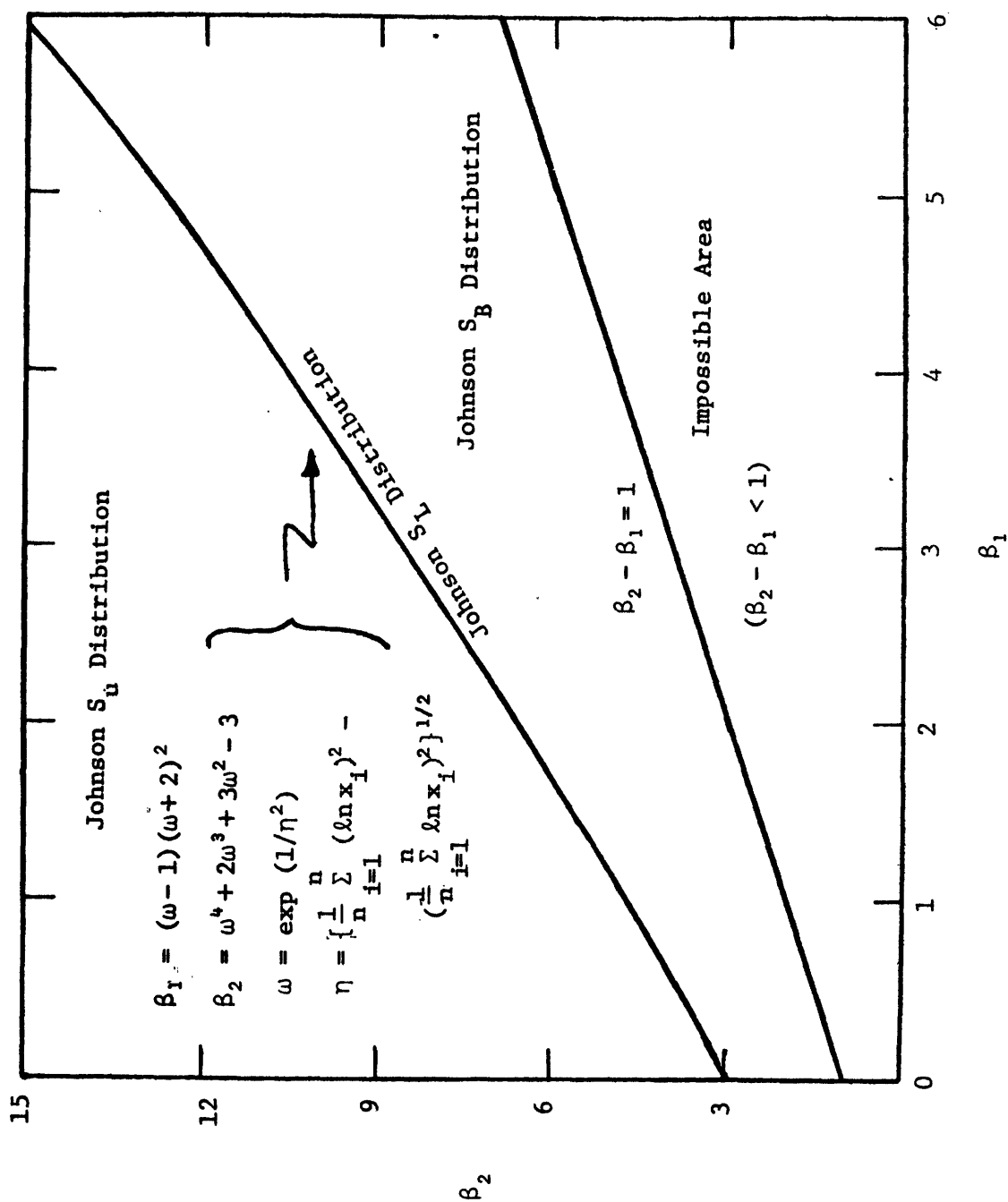


Fig. 3.18: Chart for determining appropriate Johnson distribution approximation (Hahn & Shapiro (1967)).

2001

2002

2003

2004

2005

2006

2007

2008

2009

2010

2011

2012

2013

2014

2015

2016

2017

2018

2019

CHAPTER 4: APPLICATION AND RESULTS

4.1 Introduction

In this chapter we present a series of examples to validate the effectiveness of the mathematical models developed in Chapter 2 which have been encoded in UNRAC. As stated earlier, (See Chapter 3), the UNRAC code consists of three different parts, namely, cut set generator, unreliability evaluator, and Monte-Carlo simulator. In Section 3.2 the effectiveness of the cut set generator part was discussed by comparing the UNRAC results with other published data. In order to validate other parts and the models encoded, several systems have been analyzed using UNRAC code and the results have been compared with the other codes and/or published data. They are:

- 1) Auxiliary Feed Water System (AFWS) an example from WASH 1400.
- 2) An example of an electrical system
- 3) A Chemical and Volume Control System (CVCS)
- 4) A High Pressure Injection System (HPIS), an example given in WASH 1400.

4.2 Auxiliary Feed Water System (AFWS), A Comparison with WASH 1400

The function of AFWS is to provide feed water to the secondary side of the steam generators upon loss of main feed water. In this typical example, the system consists of three pump trains, two condensate storage tanks and several

different types of valves. Figure 4.1 shows a simplified flow diagram of the AFWS that appeared in WASH 1400 (1975). The simplified (reduced) fault tree is shown in Fig. 4.2. Table 4.1 shows the data used in both WASH 1400 and this study.

The main objective of this example is to show the requirement of time dependent unreliability analysis for the standby systems in order to find the unreliability or unavailability per demand. In WASH 1400, the average point estimate unavailability of the components was used to find the top event (system) unavailability for a given interval. The pictorial summary of the results is shown in Fig. 4.3.

For this study, Fig. 4.2 was further reduced by using the techniques mentioned in Section 3.2.2 in order to eliminate the unnecessary evaluations. The fault tree used to input to UNRAC is shown in Fig. 4.4. UNRAC generated 145 cut sets and evaluated the average and time dependent unreliability of the AFWS for the interval of zero to one year. The results showed that the unavailability per demand of the AFWS (average unreliability over 1 year) was 3.114×10^{-4} *. The maximum unreliability which caused by inspection period was 4.18×10^{-2} and the minimum unreliability resulted after each repair was 5.786×10^{-5} . Figure 4.5 shows the time dependent unreliability of the system over the interval of 60 days. As can be seen, the unreliability of the system is

*NOTE: The exact results were given here for possible future check.

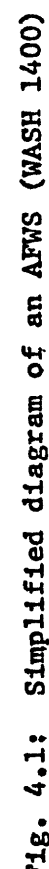


Fig. 4.1: Simplified diagram of an AFWS (WASH 1400)

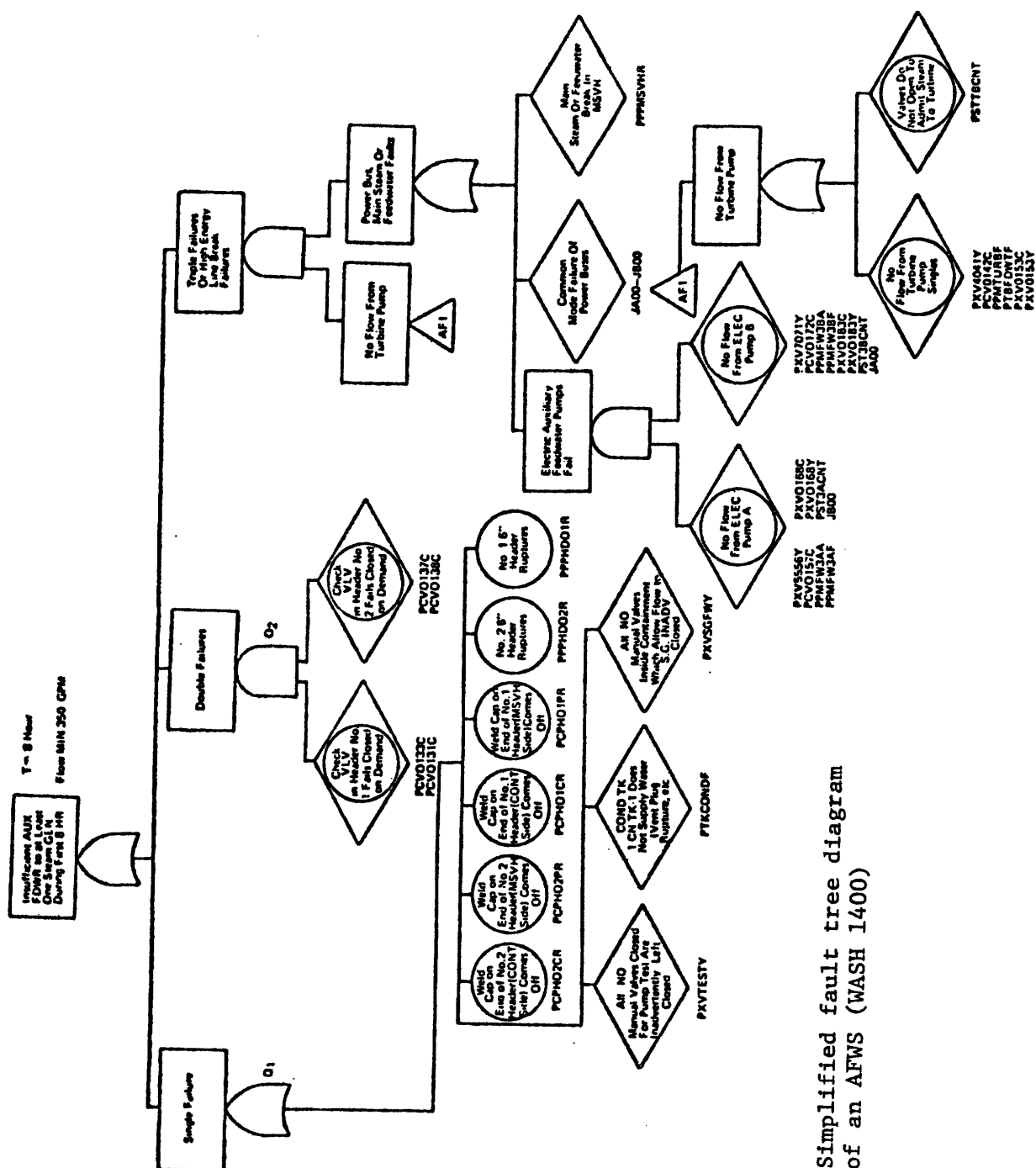


Fig. 4.2: Simplified fault tree diagram of an AFWS (WASH 1400)

[illegible]

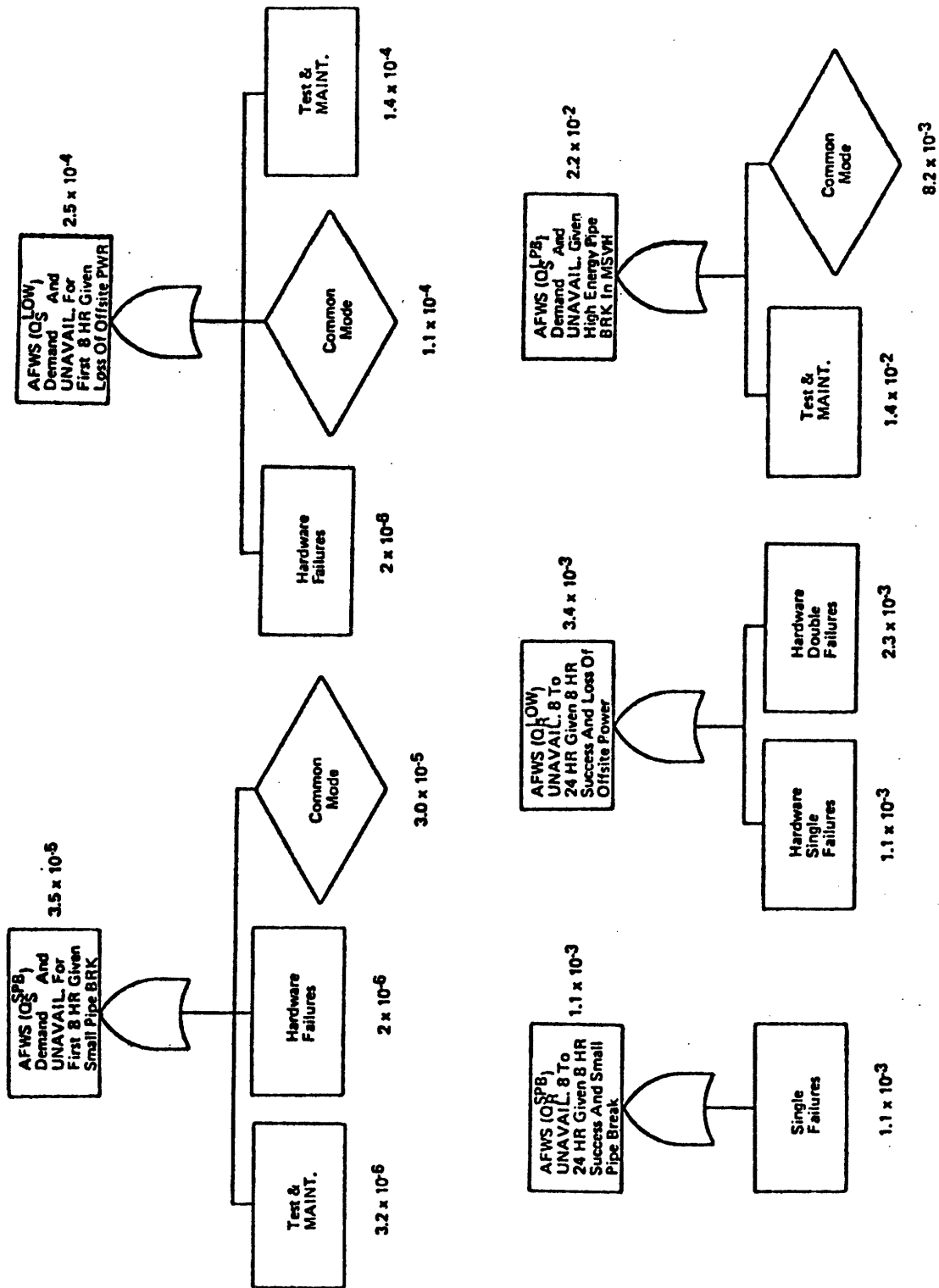


Fig. 4.3: WASH 1400 qualitative results of an AFWS pictorial summary.

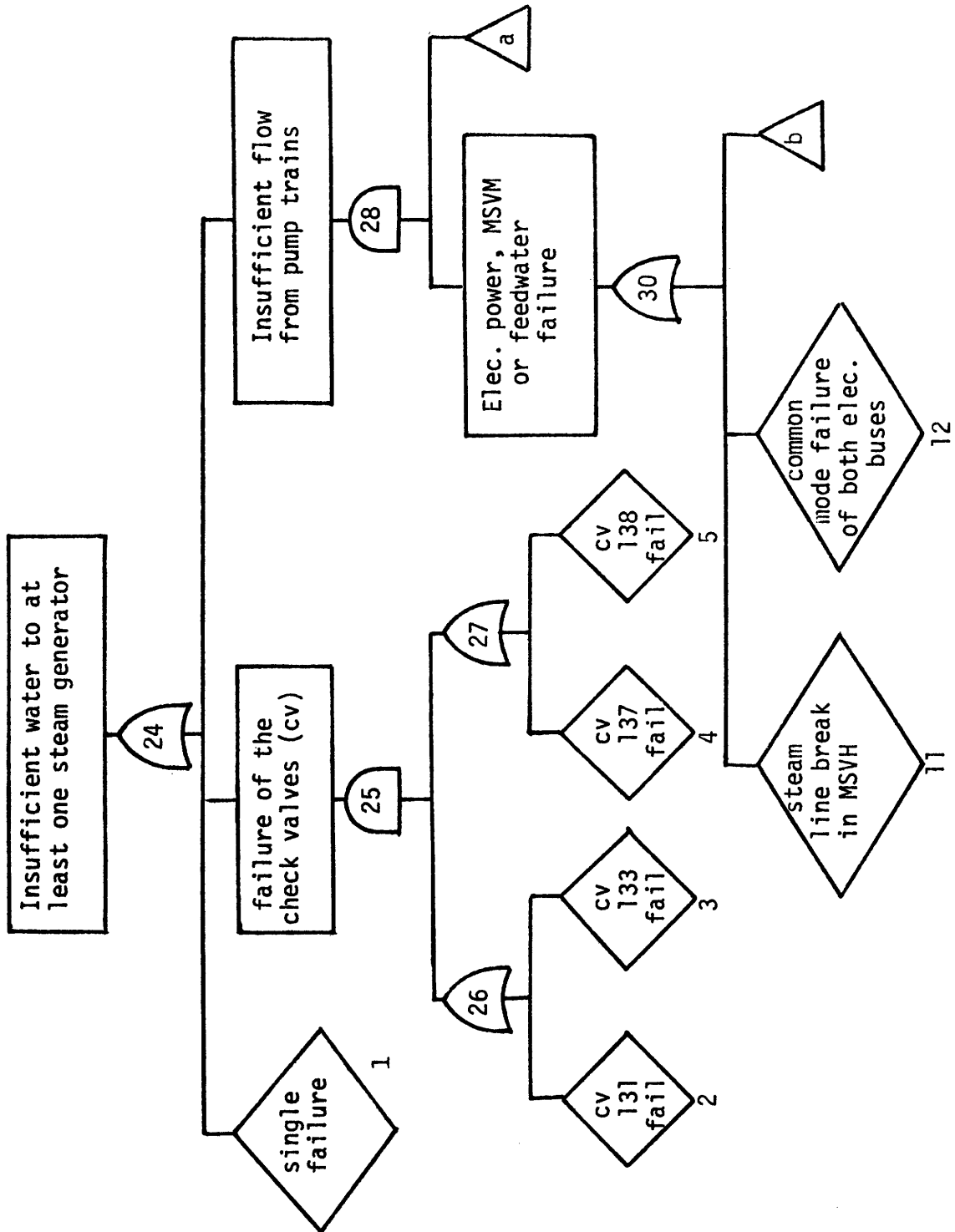


Fig. 4.4: A further reduced fault tree of AFWS.

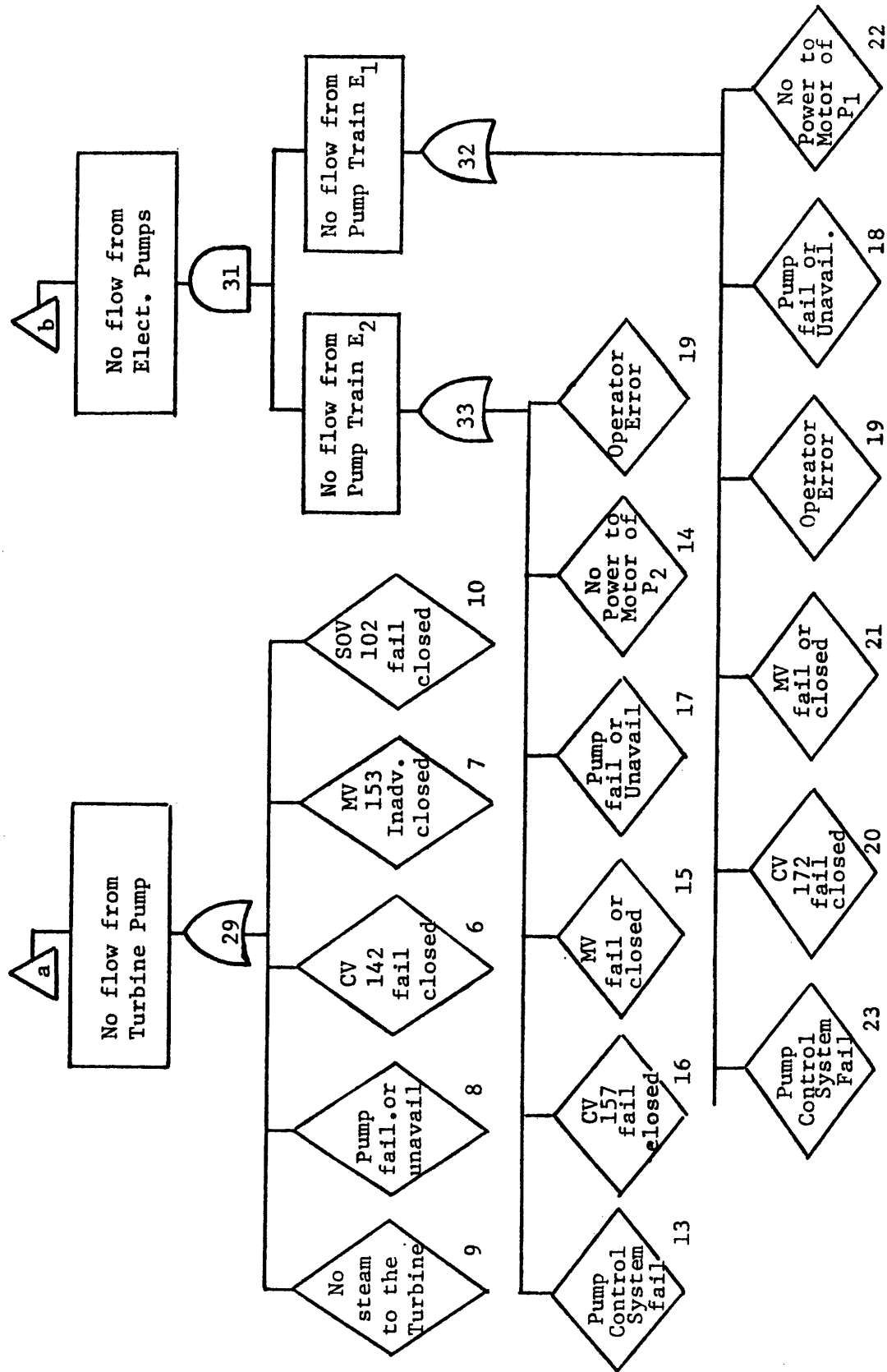


Fig. 4.4 Continued

Table 4.1: AUX-FEED WATER SYSTEM COMPONENT
FAILURE CHARACTERISTICS DATA

COMPONENT NO,	FAILURE RATE (PER HR)	TEST INTERVAL (DAYS)	FIRST INTERVAL (DAYS)	TIME FOR TESTING (HRS)	TIME FOR REPAIR (HRS)	OVERRIDE UNAVAIL	RESIDUAL UNAVAIL
1	0.0	0.0	0.0	0.0	0.0	0.0	5.10E-07
2	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04
3	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04
4	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04
5	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04
6	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04
7	3.00E-07	3.00E+01	1.00E+01	1.50E+00	4.20E+00	1.00E+00	1.00E-03
8	3.00E-05	3.00E+01	1.00E+01	1.50E+00	4.20E+00	1.00E+00	1.00E-03
9	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-03
10	1.00E-07	3.00E+01	1.00E+01	1.50E+00	4.20E+00	1.00E+00	1.00E-03
11	0.0	0.0	0.0	0.0	0.0	0.0	7.50E-05
12	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-02
13	5.10E-06	3.00E+01	2.00E+01	1.50E+00	4.20E+00	1.00E+00	2.00E-03
14	0.0	0.0	0.0	0.0	0.0	0.0	3.70E-02
15	3.00E-07	3.00E+01	2.00E+01	1.50E+00	4.20E+00	1.00E+00	1.00E-03
16	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04
17	3.00E-05	3.00E+01	2.00E+01	1.50E+00	4.20E+00	1.00E+00	1.00E-03
18	3.00E-05	3.00E+01	3.00E+01	1.50E+00	4.20E+00	1.00E+00	1.00E-03
19	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-03
20	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04
21	3.00E-07	3.00E+01	3.00E+01	1.50E+00	4.20E+00	1.00E+00	1.00E-03
22	0.0	0.0	0.0	0.0	0.0	0.0	3.70E-02
23	5.10E-06	3.00E+01	3.00E+01	1.50E+00	4.20E+00	1.00E+00	2.00E-03

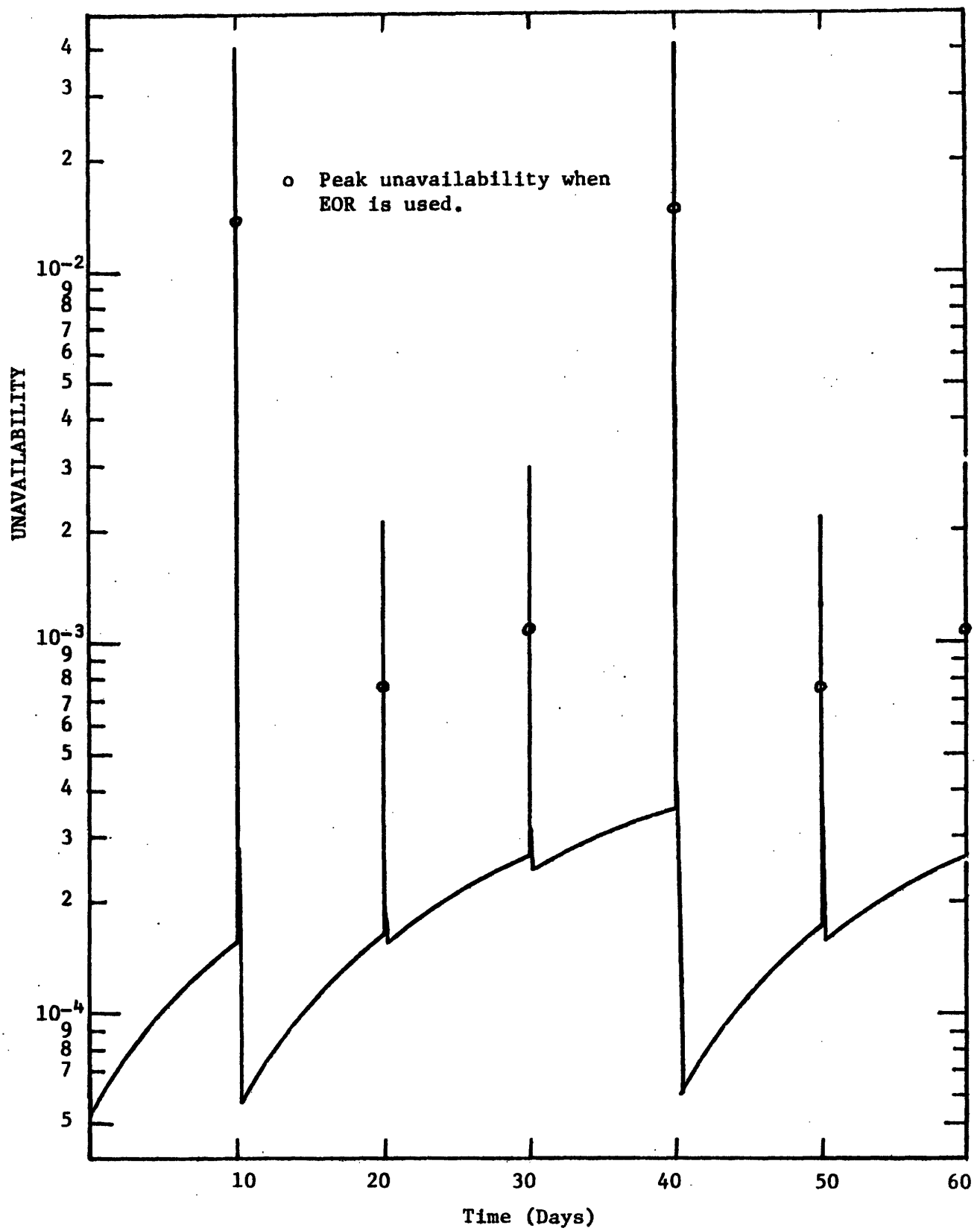


Fig. 4.5: AFWS time dependent unavailabilities as calculated by UNRAC.

highly time dependent, and varies by 3 order of magnitude. This strong time dependence of the unreliability is not revealed in WASH 1400 where only the average value is reported. This additional information can be useful in providing further insight into accidental analysis. The UNRAC code is also able to evaluate the importance of each component in producing the top event. The code determines importance by two methods: (1) Birnbaum's measure, and (2) Fussell-Vesely measure. Birnbaum's measure is the simple differentiating method and is evaluated by the following equation:

$$(B.I.)_i = \frac{\partial Q(t)}{\partial q_i(t)} = Q(t) \Big|_{q_i(t)=1} - Q(t) \Big|_{q_i(t)=0} \quad (4.1)$$

where

$Q(t)$ = top event unreliability equation, given in Eqn. (3.12),

$q_i(t)$ = unreliability of component i ,

$(B.I.)_i$ = Birnbaum's importance of component i .

whereas the Fussell-Vesely measure of importance is defined by:

$$(F.V.I.)_i = \frac{\sum \text{Prob. (cut sets containing component } i)}{Q(t)} \quad (4.2)$$

The reader should note that these two equations determine different measures of importance. The Fussell-Vesely determines the criticality importance whereas the Birnbaum's determines the actual importance of the component (Lambert (1975)). The UNRAC results for the importance analysis based on the average component's unreliability is given in Table 4.2. As would be expected, Birnbaum's measure shows that the single failure (failure of component #1) has the highest importance, whereas the Fussell-Vesely measure shows that the common power failure to both electrical pumps (failure of component #12) has the highest contributions (i.e., the component which contributes most to the top event).

To find the top event uncertainty bound and the approximated distribution, three different runs were made. For the first run we used the average point estimate unavailability of each component with the error factors on these estimates given in Table 4.3. The assumed distribution for each component involved was considered to be a log-normal. The first two columns of Table 4.4 shows the top event unavailabilities with its respective confidence levels. The error factor, which is calculated by the following equation, was found to be 3.79.

$$EF = \max \left(\frac{x_{.95}}{x_{.50}}, \frac{x_{.50}}{x_{0.05}} \right) \quad (4.3)$$

where

x_i is the value of top event unavailability with respect

Table 4.2: COMPONENT'S IMPORTANCE CALCULATIONS
OF AFWS USING FUSELL-VESELY AND
BIRNBAUM MEASURES

COMPONENT NO.	FUSSELL-VESELY MEASURE	BIRNBAUM'S MEASURE
1	1.6192E-03	1.0000E+00
2	6.3498E-05	2.0000E-04
3	6.3498E-05	2.0000E-04
4	6.3498E-05	2.0000E-04
5	6.3498E-05	2.0000E-04
6	4.6607E-03	1.4680E-02
7	1.5392E-01	1.4680E-02
8	6.4245E-01	1.4680E-02
9	4.6607E-02	1.4680E-02
10	1.5061E-01	1.4680E-02
11	5.1001E-03	2.1419E-02
12	6.8001E-01	2.1419E-02
13	2.4262E-02	1.2867E-03
14	1.5115E-01	1.2867E-03
15	1.3235E-02	1.2867E-03
16	4.0850E-04	1.2867E-03
17	5.6086E-02	1.2867E-03
18	5.6619E-02	1.2853E-03
19	6.8001E-02	2.1419E-02
20	4.0807E-04	1.2853E-03
21	1.2988E-02	1.2853E-03
22	1.5098E-01	1.2853E-03
23	2.4139E-02	1.2853E-03

Table 4.3: AUX-FEED WATER COMPONENT'S
AVERAGE UNAVAILABILITY ERROR FACTOR (EF)

<u>Component No.</u>	<u>EF on Average Unavailability</u>	<u>Component's Unavail- ability Distribution</u>
1	30.0	L*
2	3.0	L
3	3.0	L
4	3.0	L
5	3.0	L
6	3.0	L
7	6.0	L
8	3.0	L
9	10.0	L
10	3.0	L
11	30.0	L
12	3.0	L
13	3.0	L
14	3.0	L
15	6.0	L
16	3.0	L
17	10.0	L
18	10.0	L
19	10.0	L
20	3.0	L
21	6.0	L
22	10.0	L
23	3.0	L

*L stands for Log normal

Table 4.4: AUX-FEED WATER SYSTEM CUMULATIVE UNAVAILABILITY DISTRIBUTION

Cumulative Dist. Percentile	Calculated Top Event Value	Approximated Top Event Value by Log-normal, by Johnson Sg*		Best Fitted Dist. Value*
		by Log-normal	by Johnson Sg*	
0.5	9.5501E-05	7.4845E-05	9.0870E-05	1.2165E-04
1.0	1.1060E-04	9.1660E-05	1.1002E-05	1.3551E-04
5.0	1.8517E-04	1.5903E-04	1.8511E-04	1.9281E-04
10.0	2.4168E-04	2.1348E-04	2.4443E-04	2.4061E-04
15.0	2.8301E-04	2.6039E-04	2.9484E-04	2.8256E-04
20.0	3.2607E-04	3.0496E-04	3.4227E-04	3.2298E-04
30.0	4.1029E-04	3.9431E-04	4.3622E-04	4.0534E-04
40.0	5.1030E-04	4.9071E-04	5.3627E-04	4.9585E-04
50.0	6.0310E-04	6.0310E-04	6.5144E-04	6.0309E-04
60.0	7.3721E-04	7.4123E-04	7.9138E-04	7.3702E-02
70.0	9.3150E-04	9.2244E-04	9.7274E-03	9.1560E-02
80.0	1.1765E-03	1.1927E-03	1.2395E-03	1.1868E-03
85.0	1.3561E-03	1.3968E-03	1.4387E-03	1.3950E-03
90.0	1.6619E-03	1.7038E-03	1.7350E-03	1.7120E-03
95.0	2.2905E-03	2.2872E-03	2.2899E-03	2.2612E-03
99.0	4.2911E-03	3.9683E-03	3.8468E-03	4.1533E-03
99.5	5.5010E-03	4.8595E-03	4.6540E-03	5.1462E-03

*For distribution see the text

to ith confidence level.

To approximate the top event distribution, the method discussed in Section 3.6 was used. By using Fig. 3.18 and the values of β_1 and β_2 which were 11.5 and 20.3 respectively, the top event distribution is bound to be a Johnson S_B distribution with the following general equation (Hahn and Shapiro (1967)):

$$f(x) = \frac{\eta}{\sqrt{2\pi}} \frac{\lambda}{(x-\epsilon)(\lambda-x+\epsilon)} \exp\left\{-\frac{1}{2}\left[\gamma + \eta \ln\left(\frac{x-\epsilon}{\lambda-x+\epsilon}\right)\right]^2\right\} \quad (4.4)$$

$$\epsilon \leq x \leq \epsilon + \lambda, \quad -\infty < \gamma < +\infty, \quad \lambda > 0, \quad -\infty < \epsilon < +\infty$$

Since x is the unreliability value, it is bounded between zero and one, hence; $x=1$ and $\epsilon=0$. To find the other parameters, γ and η , the following set of equations were used:

$$\hat{\eta} = \frac{Z_\alpha - Z_{1-\alpha}}{\ln\left[\frac{(x_\alpha - \epsilon)(\lambda + \epsilon - x_{1-\alpha})}{(x_{1-\alpha} - \epsilon)(\lambda + \epsilon - x_\alpha)}\right]} \quad (4.5)$$

$$\hat{\gamma} = Z_\alpha - \hat{\eta} \ln \frac{x_\alpha - \epsilon}{\lambda + \epsilon - x_\alpha}$$

where α = percentile $> .50$

Z_α = is the α 100th percentile for a standard normal variate

$\hat{\eta}, \hat{\gamma}$ = are the estimated values for η and γ

x_α = is the α 100th percentile for the calculated data

Substituting for ϵ , λ , η and γ in Eqn. (4.4) we get:

$$f(y) dy = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}y^2} dy, \quad (4.6)$$

$$-\infty < y < +\infty$$

where:

$$y = \hat{\gamma} + \hat{\eta} \ln \frac{x}{1-x}$$

Column 4 of the Table 4.4 also summarizes the estimated values of top event unavailability using approximated distribution given by Eqn. (4.6) with $\hat{\eta}$ and $\hat{\gamma}$ evaluated at $\alpha = 0.95$. Column 3 of Table 4.4 gives the top event distribution assuming a log-normal distribution with the median of 6.0310E-04 and an error factor of 3.97. The latter approximation to the top event distribution was used in WASH 1400. Figure 4.6 clarifies the difference between the actual results and the above approximation distributions results. As can be seen, for the majority of the data the simple log-normal distribution can approximate the top event distribution very well. This figure supports the validity of the WASH 1400 assumption of log-normal distribution for the whole system. However, the distribution that best fits the top event is found to be the S_B Johnson distribution with the following parameters: $\hat{\lambda} = 1$, $\hat{\epsilon} = 6.48730E-5$, $\hat{\eta} = 1.1446445$, $\hat{\gamma} = 8.6154013$.

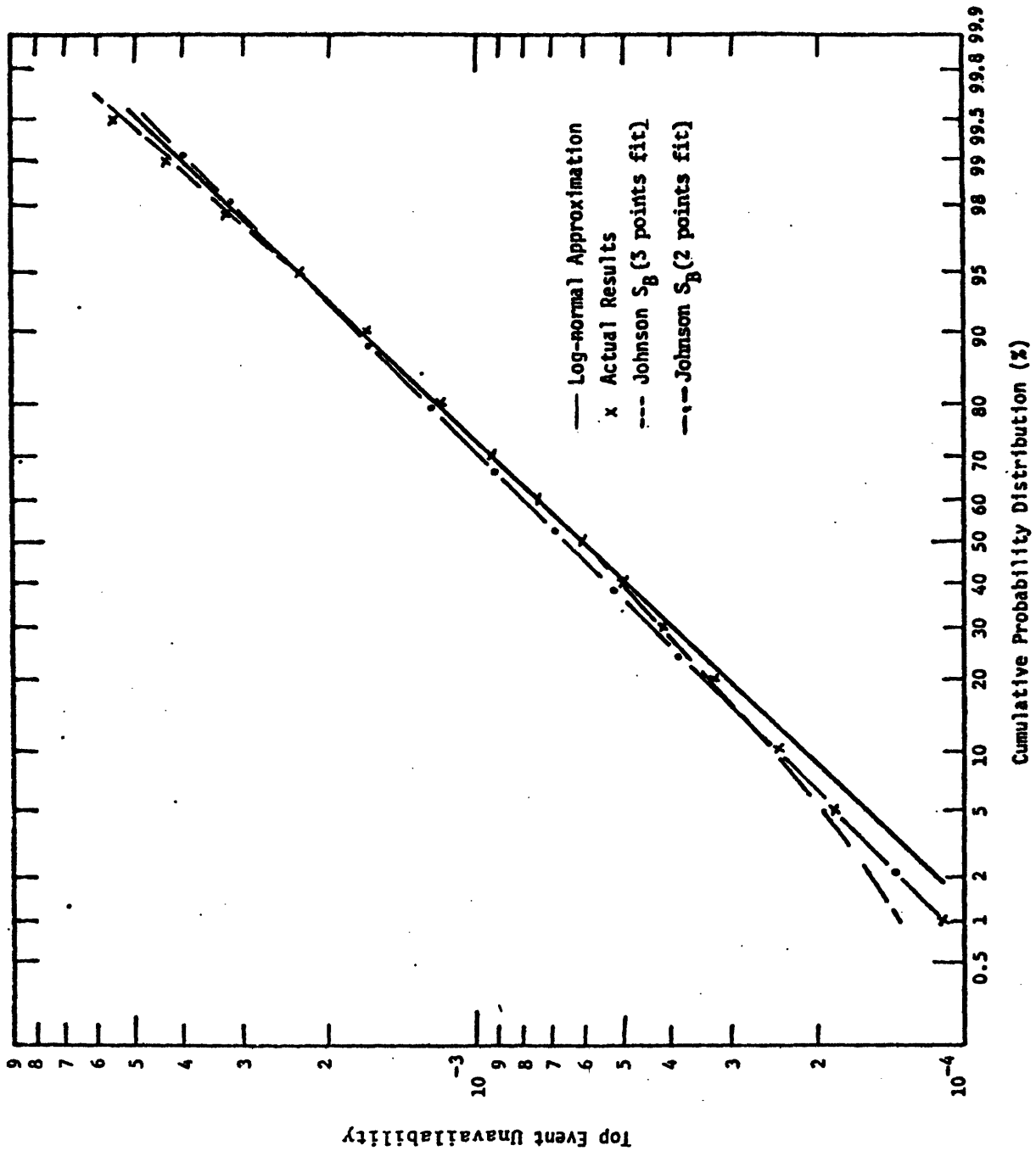


Fig. 4.6: Cumulative top event distribution of AFWS, evaluated based on the average unavailability of each component.

The above values were found by matching 25th, 50th, and 75th percentile of top event with the fitted distribution (cf. Eqn. (4.4)). It is worth mentioning that the value of $\hat{\epsilon}$ was found by the following equation:

$$\frac{(x_{0.5} - \hat{\epsilon})(\hat{\epsilon} + \lambda - x_{1-\alpha})}{(\hat{\epsilon} + \lambda - x_{0.5})(x_{1-\alpha} - \hat{\epsilon})} = \frac{(x_{\alpha} - \hat{\epsilon})(\lambda + \hat{\epsilon} - x_{0.5})}{(\hat{\epsilon} + \lambda - x_{\alpha})(x_{0.5} - \hat{\epsilon})} \quad (4.7)$$

where

λ = assumed to be 1

α = is the same parameter as in Eqn. (4.5)

For the second run we used error factors on the individual component's characteristics (i.e., failure rate, repair rate, etc.). Table 4.5 shows the data used to evaluate the top event uncertainty bound. The error factor of the top event was found to be 3.654. The evaluation was carried out by simulating each component's characteristics distribution with Monte Carlo sampling and finding the average unavailability of each component at every iteration to calculate the top event values. Again the simple log-normal distribution seems to approximate the top event quite well (See Fig. 4.7).

For the third run, first the error factor on some of the component's parameters data were changed to see the effect of the longer tail for log-normal distribution on the top event.

Table 4.5: AFWS COMPONENT FAILURE CHARACTERISTICS
ERROR FACTORS DATA

COMPONENT NO.	LAMDA	DIS	TC	DIS	TR	DIS	QRESID	DIS
1	0.0		0.0		0.0		30.000	L
2	0.0		0.0		0.0		3.000	L
3	0.0		0.0		0.0		3.000	L
4	0.0		0.0		0.0		3.000	L
5	0.0		0.0		0.0		3.000	L
6	0.0		0.0		0.0		3.000	L
7	3.000	L	3.000	L	3.000	L	6.000	L
8	3.000	L	3.000	L	3.000	L	3.000	L
9	0.0		0.0		0.0		10.000	L
10	3.000	L	3.000	L	3.000	L	3.000	L
11	0.0		0.0		0.0		30.000	L
12	0.0		0.0		0.0		3.000	L
13	3.000	L	3.000	L	3.000	L	3.000	L
14	0.0		0.0		0.0		3.000	L
15	3.000	L	3.000	L	3.000	L	6.000	L
16	0.0		0.0		0.0		3.000	L
17	10.000	L	3.000	L	3.000	L	3.000	L
18	10.000	L	3.000	L	3.000	L	3.000	L
19	0.0		0.0		0.0		10.000	L
20	0.0		0.0		0.0		3.000	L
21	3.000	L	3.000	L	3.000	L	3.000	L
22	0.0		0.0		0.0		10.000	L
23	3.000	L	3.000	L	3.000	L	3.000	L

The values in columns 2,4,6 and 8 are the error factors on failure rate, average test time, average repair time and residual unavailability of each component, respectively.

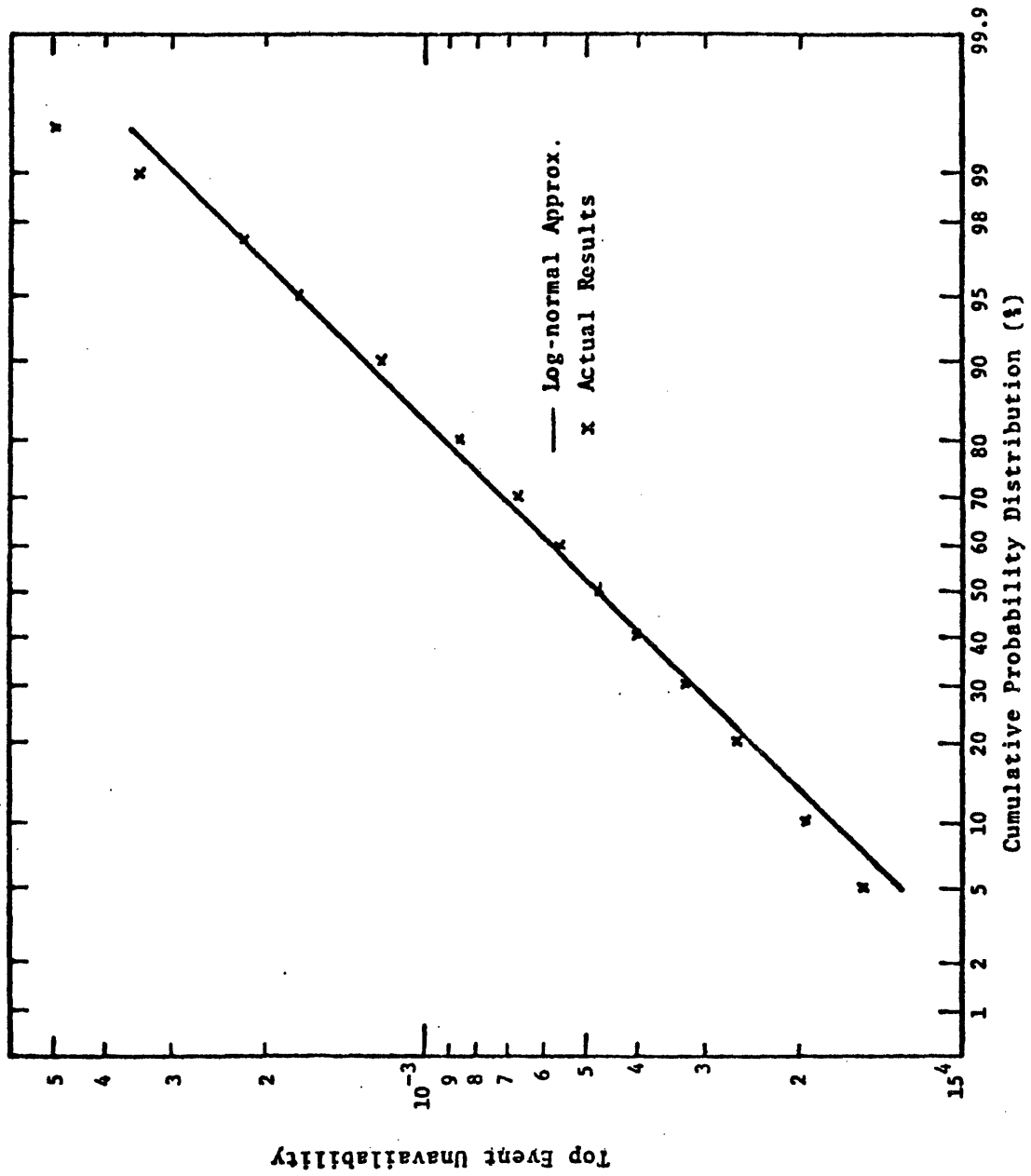


Fig. 4.7: AFWS top event cumulative distribution for mixed component's distribution.

Second, it was assumed that the mean test and repair time follow a gamma distribution. Then the top event unreliability was evaluated for 1200 and 5000 Monte-Carlo iterations for both cases. Table 4.6 summarizes the input error factors for this run, and Table 4.7 and 4.8 give the top event unreliability computations. The results show although there exist differences between the values of the top event unreliability for 1200 and 5000 iterations, the 1200 iterations will suffice for the analysis.

In another run, the importance of the components in the system were evaluated based on the assumption that electric pump number 1 is down, i.e., under the test at 721 hours. Based on the above assumption and knowing that when one pump is down the components in that pump train are ineffective for delivering water, the UNRAC regenerates the new minimal cut set by discarding all the failed and ineffective components. This capability is provided in order to make the code more versatile in importance analysis while preserving other features of the code and, in addition, the original system fault tree is only required input to the code for all the subsequent evaluations. The code internally changes the system configuration by the relevant data that user will supply as input for the intended job. The results of the importance analysis at 721 hours of operation is shown in Table 4.9. As can be seen, at this time of operation, the turbo-pump is the dominant component in the top event

Table 4.6: AFWS COMPONENT FAILURE CHARACTERISTICS
ERROR FACTOR DATA, AN EXAMPLE FOR LONG
TAIL LOG-NORMAL AND GAMMA DISTRIBUTIONS

COMPONENT NO.	LAMDA	DIS	TC	DIS *	TR	DIS *	QRESID	DIS
1	0.0		0.0		0.0		30.000	L
2	0.0		0.0		0.0		10.000	L
3	0.0		0.0		0.0		10.000	L
4	0.0		0.0		0.0		10.000	L
5	0.0		0.0		0.0		10.000	L
6	0.0		0.0		0.0		10.000	L
7	10.000	L	3.000	L	3.000	L	10.000	L
8	10.000	L	3.000	L	3.000	L	10.000	L
9	0.0		0.0		0.0		10.000	L
10	10.000	L	3.000	L	3.000	L	10.000	L
11	0.0		0.0		0.0		30.000	L
12	0.0		0.0		0.0		3.000	L
13	10.000	L	3.000	L	3.000	L	10.000	L
14	0.0		0.0		0.0		10.000	L
15	10.000	L	3.000	L	3.000	L	10.000	L
16	0.0		0.0		0.0		10.000	L
17	10.000	L	3.000	L	3.000	L	10.000	L
18	10.000	L	3.000	L	3.000	L	10.000	L
19	0.0		0.0		0.0		10.000	L
20	0.0		0.0		0.0		10.000	L
21	10.000	L	3.000	L	3.000	L	10.000	L
22	0.0		0.0		0.0		10.000	L
23	10.000	L	3.000	L	3.000	L	10.000	L

* T_C and T_R distributions are changed to gamma distribution for calculating the values in Table 4.8.

**Table 4.7: THE EFFECT OF THE MONTE-CARLO
SIMULATION TRIALS ON THE TOP
EVENT DISTRIBUTION**

Level of	Simulation Trials	
	1200	5000
CONFIDENCE	TOP EVENT VAL	TOP EVENT VAL
0.5%	7.5631E-05	6.8032E-05
1.0%	8.6380E-05	8.1885E-05
2.5%	1.0510E-04	1.0826E-04
5.0%	1.3432E-04	1.3495E-04
10.0%	1.8046E-04	1.8212E-04
15.0%	2.2543E-04	2.2493E-04
20.0%	2.6124E-04	2.6666E-04
25.0%	3.0705E-04	3.0928E-04
30.0%	3.6497E-04	3.5501E-04
40.0%	4.8247E-04	4.6209E-04
50.0%	5.9415E-04	5.8590E-04
60.0%	7.6224E-04	7.5284E-04
70.0%	1.0068E-03	1.0201E-03
75.0%	1.2071E-03	1.2202E-03
80.0%	1.4576E-03	1.4824E-03
85.0%	1.8216E-03	1.9242E-03
90.0%	2.5920E-03	2.6107E-03
95.0%	4.3103E-03	4.3677E-03
97.5%	5.8253E-03	6.8399E-03
99.0%	1.1847E-02	1.2950E-02
99.5%	1.9284E-02	1.9408E-02

Table 4.8: THE EFFECT OF THE MONTE-CARLO
SIMULATION TRIALS ON THE TOP
EVENT DISTRIBUTION FOR MIXED
COMPONENT'S FAILURE DISTRIBUTIONS

Level of	Simulation Trials	
	1200	5000
CONFIDENCE	TOP EVENT VAL	TOP EVENT VAL
0.5%	8.4943E-05	7.6879E-05
1.0%	9.8884E-05	8.8166E-05
2.5%	1.2106E-04	1.1801E-04
5.0%	1.5033E-04	1.4395E-04
10.0%	1.9731E-04	1.8879E-04
15.0%	2.2975E-04	2.2668E-04
20.0%	2.6790E-04	2.6786E-04
25.0%	3.1991E-04	3.1454E-04
30.0%	3.5748E-04	3.5738E-04
40.0%	4.6741E-04	4.6144E-04
50.0%	5.8465E-04	5.8858E-04
60.0%	7.8831E-04	7.6327E-04
70.0%	1.0462E-03	1.0250E-03
75.0%	1.2104E-03	1.1997E-03
80.0%	1.4565E-03	1.4865E-03
85.0%	1.7495E-03	1.8652E-03
90.0%	2.4342E-03	2.5816E-03
95.0%	4.2475E-03	4.2737E-03
97.5%	6.5750E-03	6.9437E-03
99.0%	1.2499E-02	1.2298E-02
99.5%	1.6695E-02	1.6695E-02

Table 4.9: IMPORTANCE ANALYSIS OF AFWS
COMPONENTS GIVEN THAT ONE OF
THE PUMP TRAIN IS OUT OF SERVICE

COMPONENT NO.	FUSSELL-VESELY MEASURE	BIRNBAUM'S MEASURE
1	4.5248E-04	1.0000E+00
2	1.7744E-05	2.0000E-04
3	1.7744E-05	2.0000E-04
4	1.7744E-05	2.0000E-04
5	1.7744E-05	2.0000E-04
6	5.3813E-03	6.0653E-02
7	6.1554E-02	6.0653E-02
8	8.2237E-01	6.0653E-02
9	5.3813E-02	6.0653E-02
10	5.6393E-02	6.0653E-02
11	1.2359E-03	1.8574E-02
12	1.6479E-01	1.8574E-02
13	5.3087E-02	1.8574E-02
14	6.0973E-01	1.8574E-02
15	1.7663E-02	1.8574E-02
16	1.6479E-03	1.8574E-02
17	1.3488E-01	1.8574E-02
19	1.6479E-02	1.8574E-02

unavailability with the value of $8.2237\text{E}-01$ (See Fussell-Vesely measures), where as Birnbaum's measure again predicts that the single failure has the highest importance.

For the final run in this section, the independent failure of the components in each pump train were assumed to have an Exclusive OR (EOR) gate logic. In fact, this is the realistic assumption, because when the manual valve or check valve fails in the pump train, that train will be unable to deliver water. In the Table 4.1, we have initialized 3 components of each pump train to be a periodically tested component. Therefore, in their operational history they are unavailable with the probability of 1 while under the test. Hence, for the top event evaluation using the general minimal cut sets we may overestimate the unavailability of the system. To avoid such evaluations, those OR gates that have periodically tested components should be replaced by EOR gate. The EOR gate will put the gate output to 1 when one of its input has occurred. While for the case of OR gate the output value is greater than 1. To check our system for such possibility, the independent failure of each pump train was assumed to be interconnected by an EOR gate. Then the technique mentioned to transform the EOR gate to simple AND, OR and NOT gate (See Fig. 3.3), were used. The final results showed that the peak values shown in Fig. 4.5 are overestimated by a factor of three. It is worth mentioning that, care must be taken in preparing the new fault

tree with EOR gate. For example, in the data for AFWS we scheduled the simultaneous testing for the periodically tested components in each pump train, i.e., in turbine pump train. The pump, the manual valve, and solenoid valve which open the steam, are tested each month at the same time. Now if we use the EOR gate for these three components then the output value for the time of the test will be zero instead of real value of 1. The actual value of the maximum unreliability during each testing period is also shown in Fig. 4.5 (symbolically shown by o on the figure). As stated earlier, these points are a factor 3 less.

4.3 An Example of an Electrical System, Comparison of UNRAC with FRANTIC and BIT-FRANTIC

Figure 4.8 shows a simple electrical system used by Modarres (1979). The main function of this system is to provide light when the switch is closed. Relay No. 1 is a normally open (N.O.) relay and its contacts are closed when the switch is open. Relay No. 2 is a normally closed relay and it will be deenergized (contacts will be opened) if the switch is closed. Figure 4.9 shows a fault tree of Fig. 4.8 which neglects operator error, wiring failures and secondary failures. Table 4.10 shows the arbitrarily selected data that has been used to analyze the system in Modarres (1979).

First, the system structure function was formulated and provided as input to both FRANTIC (Vesely and Goldberg (1977)) and UNRAC. These results termed "exact" for the two cases are shown in Fig. 4.10. Second, to compare the results

of UNRAC and BIT-FRANTIC (Karimi and Wolf (1978)), the fault tree was directly input and the first term approximation (rare event) (cf. Eqn. (3.13)) was used to find the top event unreliability. Figure 4.10 also summarizes the results of these runs (1st term). As can be seen, UNRAC and FRANTIC exact results are only comparable where the unreliabilities are small, i.e., less than 10^{-2} . The results of UNRAC (Exact) and UNRAC (1st term approx.) are consistent to a higher degree of unreliability than the former case. The difference between UNRAC (Exact) and UNRAC (1st term approx.) at the end of each interval are caused by the approximation used in evaluating the top event. The rare event approximation results in a conservative value when the average unavailability of basic components are on the order of 10^{-1} or higher. The results of BIT-FRANTIC (1st term approx.) are more conservative; however, it is believed that the UNRAC results are more realistic.

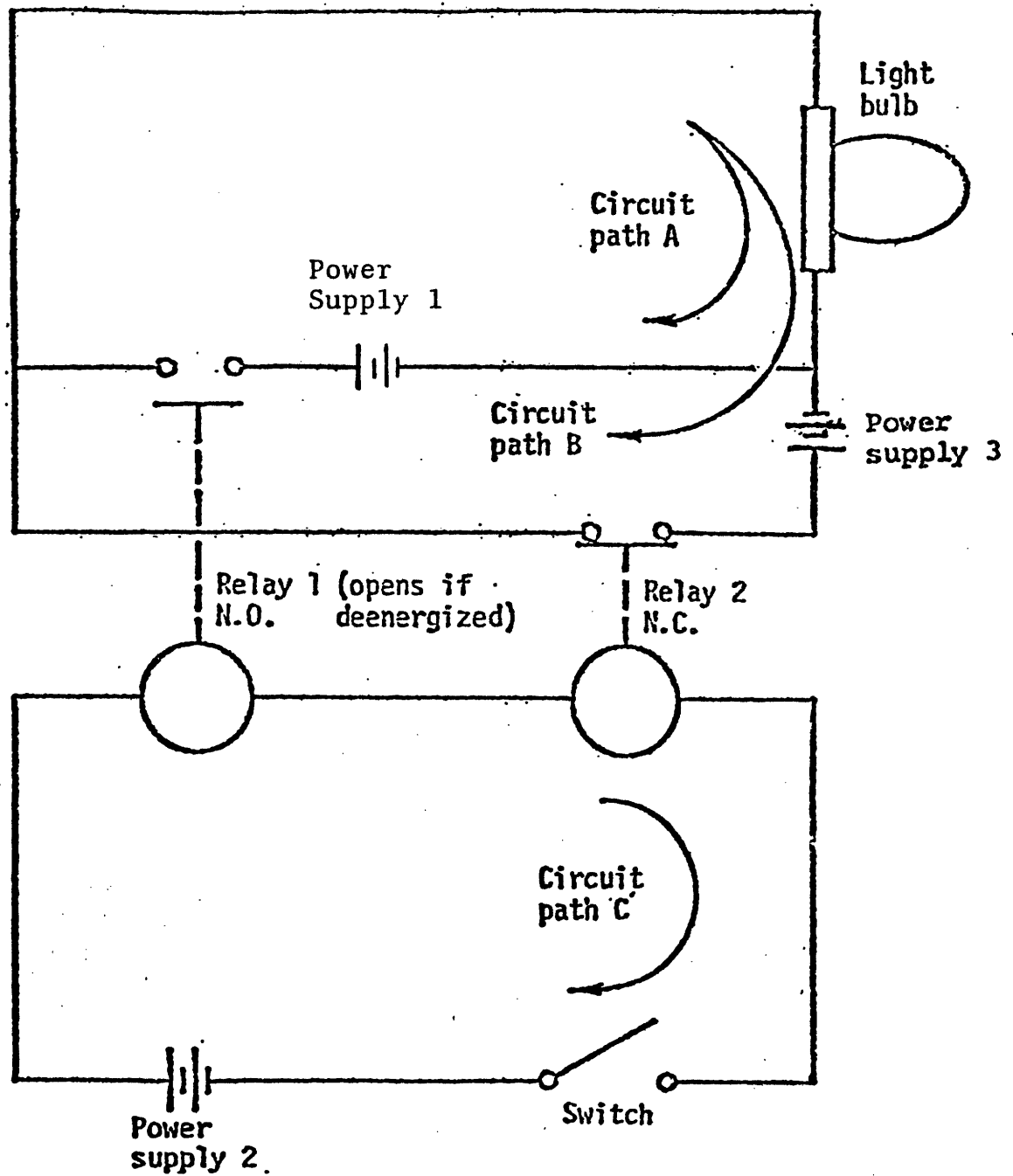


Fig 4.8: An example of an electrical system (Modarres (1979))

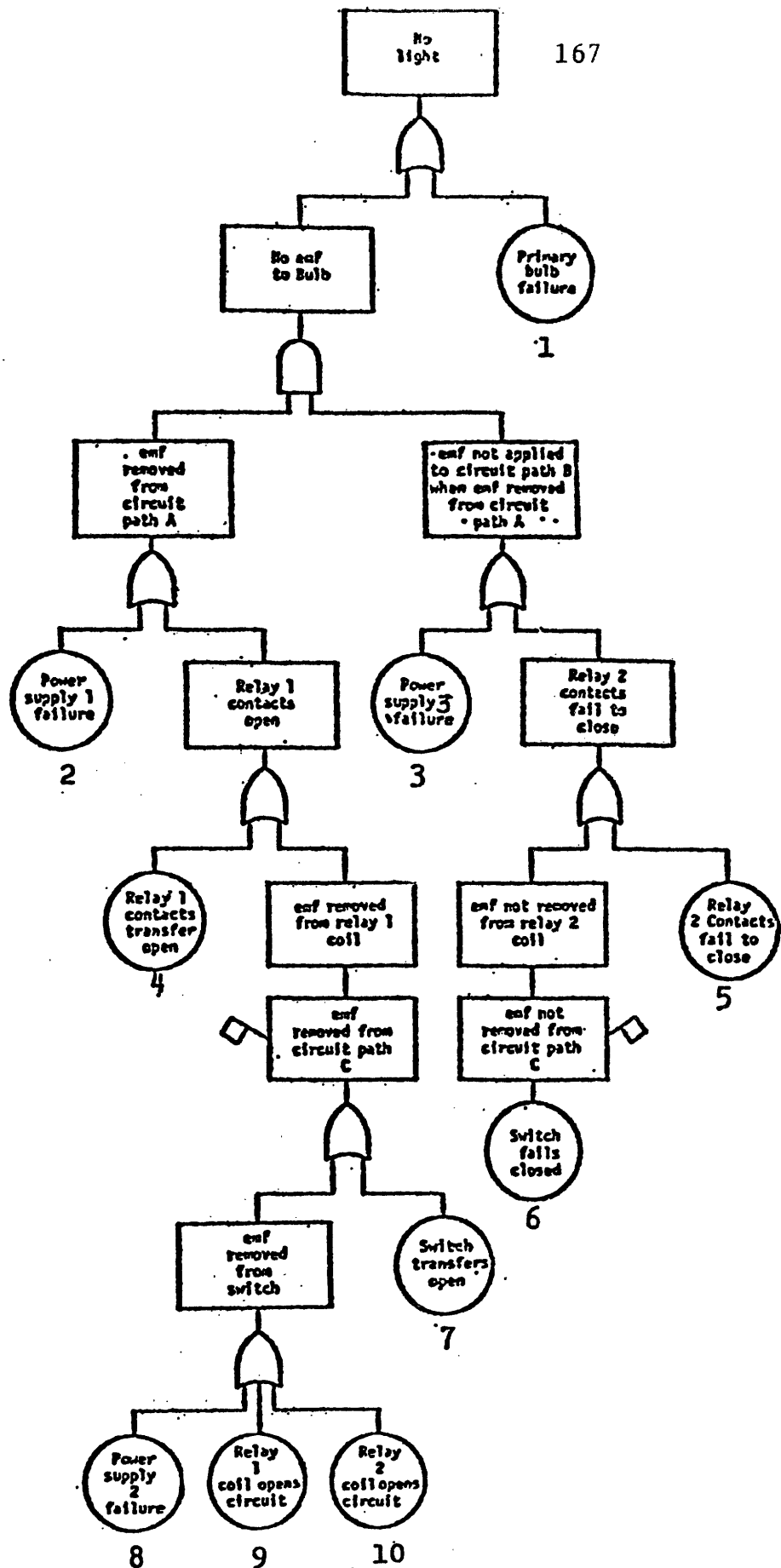


Fig 4.9: A fault tree diagram of the example electrical system

Table 4.10: COMPONENT'S PARAMETERS DATA FOR THE ELECTRICAL EXAMPLE

Component Number	Failure Rate hr^{-1}	Time Interval (days)	First Time Interval (days)	Inspection Time (hrs)	Repair Time (hrs)	Override Unavailability
1	2.0×10^{-4}	7	7	1.0	2.0	1.0
2	2.8×10^{-5}	14	7	0.5	1.0	1.0
3	2.8×10^{-5}	14	7	1.0	1.5	1.0
4	3.2×10^{-3}	7	7	0.5	2.0	1.0
5	4.1×10^{-4}	21	14	0.5	1.5	1.0
6	3.2×10^{-4}	28	28	1.0	1.5	1.0
7	2.8×10^{-3}	21	14	0.5	2.0	1.0
8	2.8×10^{-3}	14	14	1.0	1.5	1.0
9	4.5×10^{-3}	7	7	0.5	2.5	1.0
10	4.5×10^{-3}	7	7	1.5	3.0	1.0

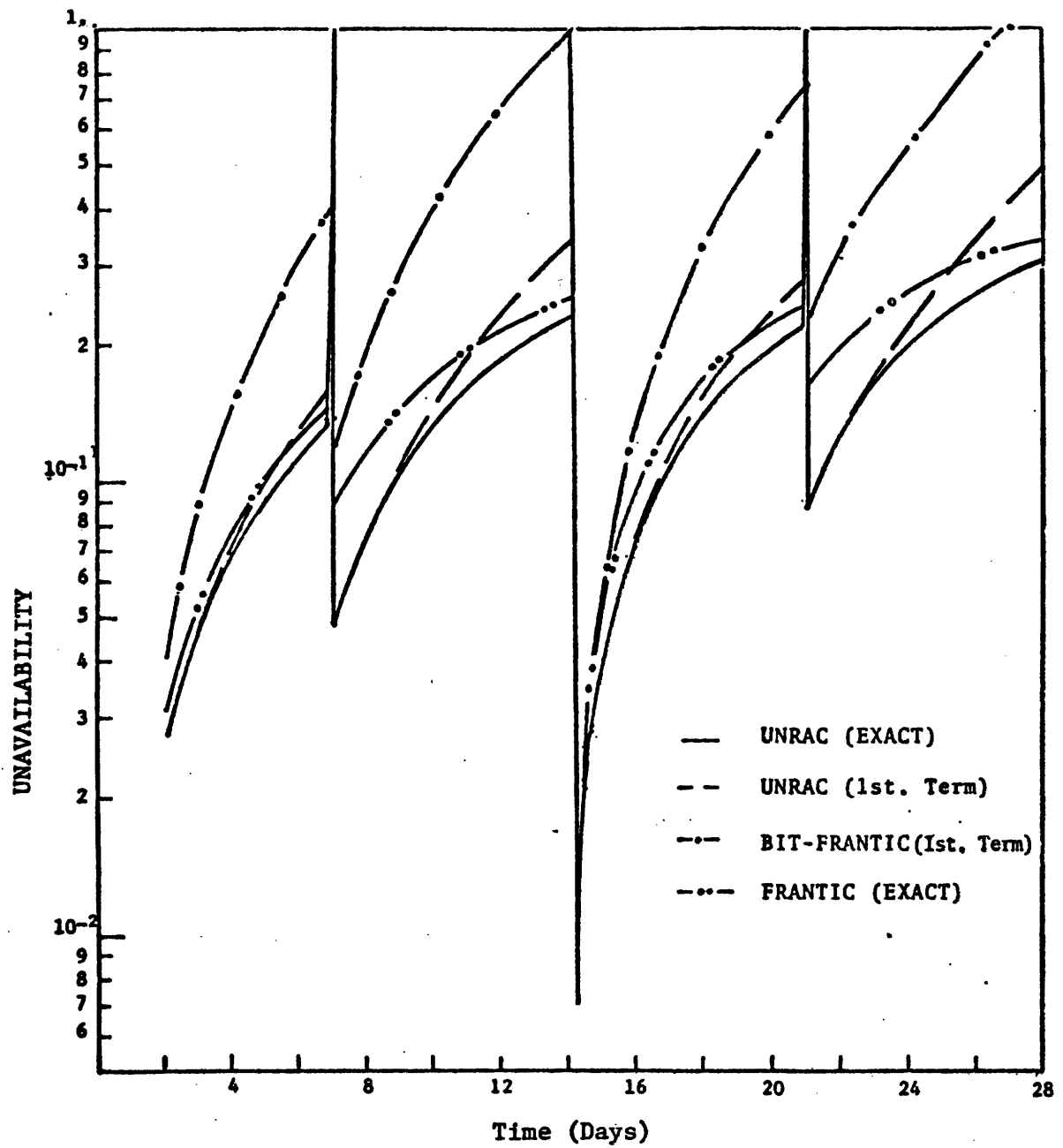


Fig. 4.10: Comparison of time dependent unavailabilities of the electrical system as calculated by UNRAC, FRANTIC and BIT-FRANTIC.

4.4 A Chemical and Volume Control System, (CVCS)

The Chemical and Volume Control System provides a means for injection of control poison in the form of boric acid solution, chemical additions for corrosion control and reactor coolant cleanup and degasification. This system also adds makeup water to the reactor coolant system (RCS), provides seal water injection to the reactor coolant pump seals, reduces the inventory of radioactivity in RCS and can be used as a high pressure injection system. The typical system under study has three pump trains which have to pump 44 gal/min. of the purified water from the volume control tank to the reactor coolant pump seals and RCS. Figure 4.11 shows the simplified pump trains of the CVCS. Each pump can only deliver a maximum of 44 gal/min. under the design condition. Therefore, we shall assume that two pumps are needed to meet the design requirement of 44 gal/min. In addition, safety regulations require that two out of three pumps be available except that technical specifications do permit two pumps to be down for a period of not more than 24 hours. Hence, in our analysis we defined the top event failure to be the simultaneous unavailability of any two pump trains. Figure 4.12 shows the simplified fault tree diagram of the system. In order to be able to compare the results of UNRAC with BIT-FRANTIC, the pumps were modelled to follow a periodically tested component. In other words we assumed that each pump undergoes a thorough test and repair after each T_2 days, and if any pump fails between the test

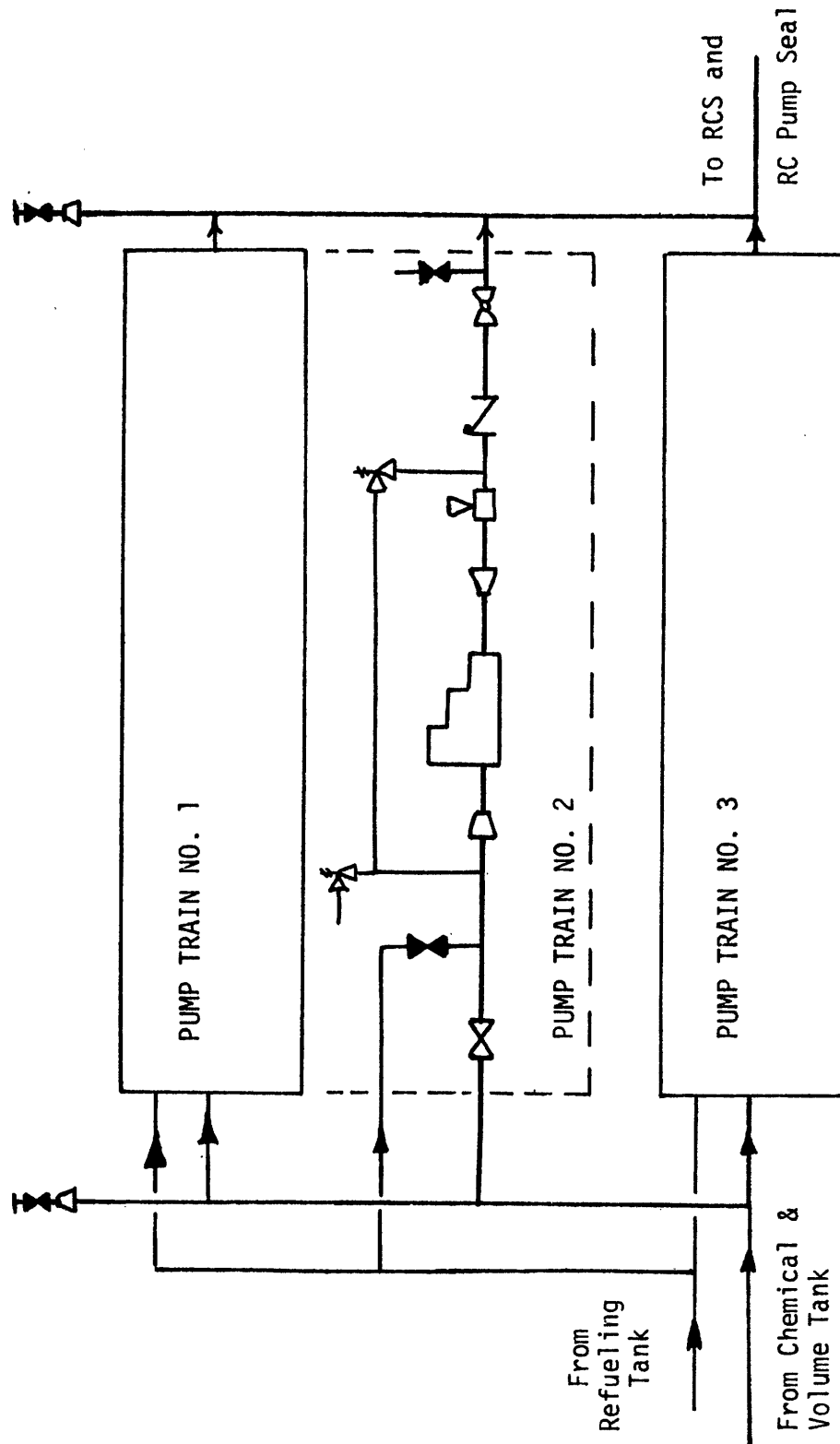


Fig. 4.11: CVCS pump trains schematic diagram.

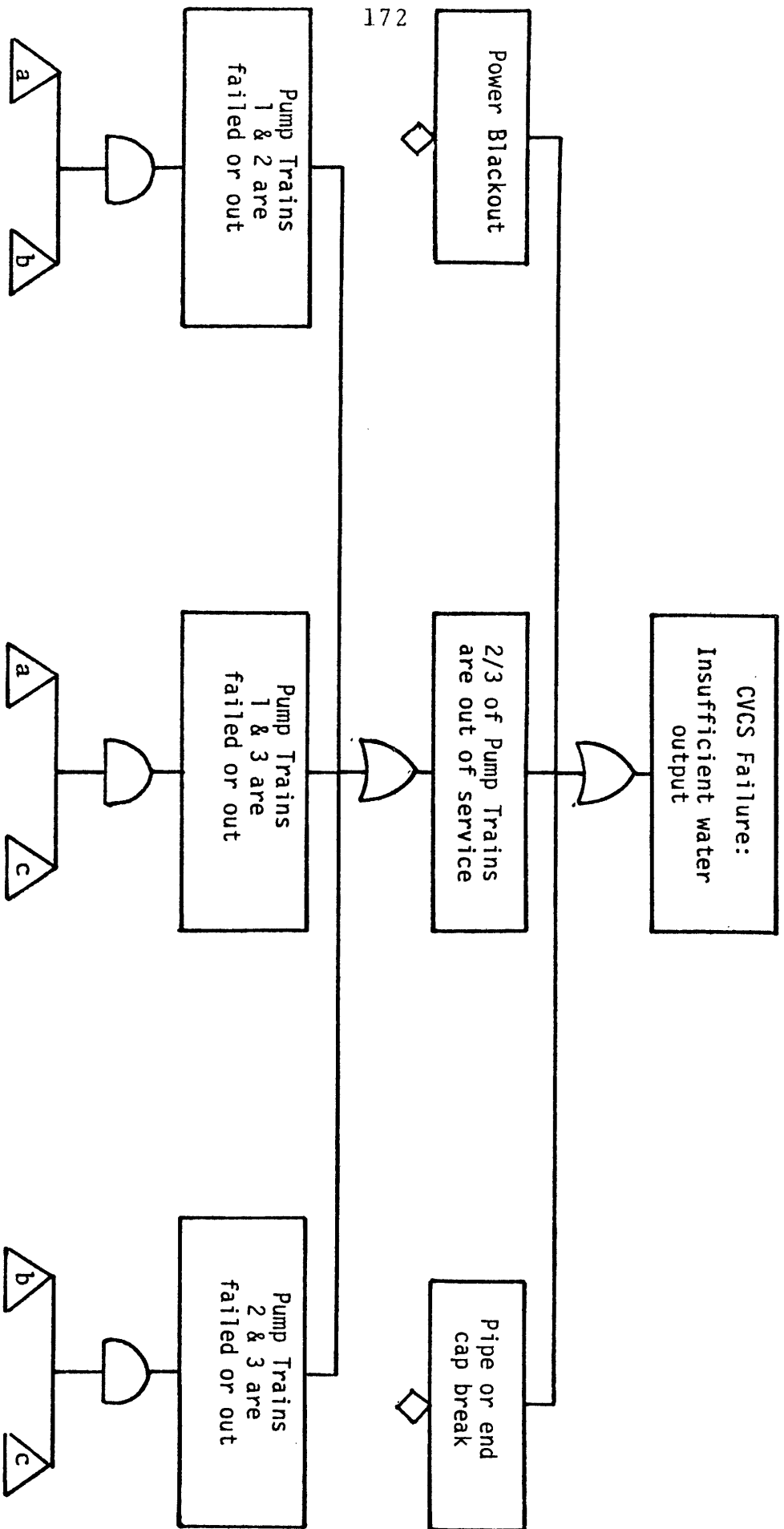


Fig. 4.12: Simplified fault tree diagram of CVCS pump system.

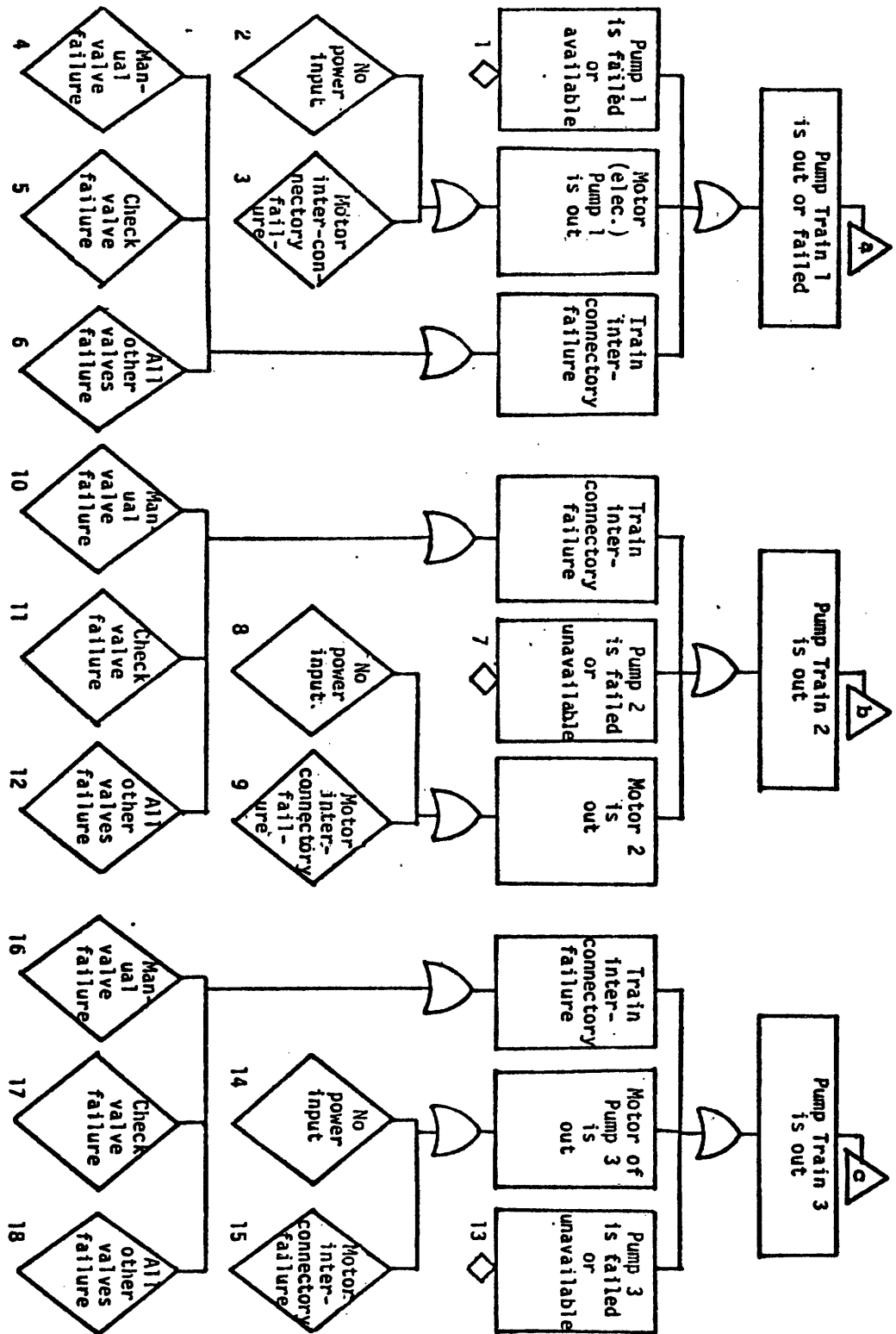


Fig. 4.12: continued.

interval it will be restored without changing its failure probability. To analyze the system under such an assumption, the data shown in Table 4.11 is used. Figure 4.13 gives the time dependent unavailability calculated by UNRAC and BIT-FRANTIC. The difference between the results of these two codes arises from the use of the rare event approximation in FRANTIC.

In another example it was assumed that the pumps are normally working during operation (not in stand-by). In this case, the pumps are unavailable if they are down because of repair or test. To analyze the system unreliability it was assumed that there is a 20% chance that the pump needs to be tested if it fails, i.e. $P_3 = 0.2$ (See Fig. 3.8) and also there is a 20% chance that the pump trip was spurious and therefore the pump can be restarted, i.e., $P_1 = 0.2$. The rest of the data given in Table 4.11 remains unchanged. The results of the top event unavailability for two types of repair distributions, namely exponential and 2nd order Erlangian are summarized in Table 4.12.

Having analyzed the CVCS pumping system by one of the above methods (either assume that the pumps are periodically maintained or usually running (monitored)), one can check the design requirement for the system availability. For example, in the first method, the unavailability per demand of CVCS pumping system is 3.08×10^{-1} or 1 out of three times the pumping system cannot meet the design requirement. If one

Table 4.11: CVCS PUMP TRAINS COMPONENT'S PARAMETERS DATA

COMPONENT NO.	FAILURE RATE (PER HR)	TEST INTERVAL (DAYS)	FIRST INTERVAL (DAYS)	TIME FOR TESTING (HRS)	TIME FOR REPAIR (HRS)	OVRRIDE UNAVAIL	RESIDUAL UNAVAIL
1	1.19E-03	3.00E+01	1.00E+01	8.00E+00	8.00E+00	1.00E+00	1.00E-03
2	0.0	0.0	0.0	0.0	0.0	0.0	5.00E-05
3	0.0	0.0	0.0	0.0	0.0	0.0	2.00E-03
4	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04
5	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04
6	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-03
7	1.19E-03	3.00E+01	2.00E+01	8.00E+00	8.00E+00	1.00E+00	1.00E-03
8	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-03
9	0.0	0.0	0.0	0.0	0.0	0.0	2.00E-03
10	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04
11	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04
12	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-03
13	1.19E-03	3.00E+01	3.00E+01	8.00E+00	8.00E+00	1.00E+00	1.00E-03
14	0.0	0.0	0.0	0.0	0.0	0.0	5.00E-05
15	0.0	0.0	0.0	0.0	0.0	0.0	2.00E-03
16	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04
17	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04
18	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-03
19	0.0	0.0	0.0	0.0	0.0	0.0	5.84E-07

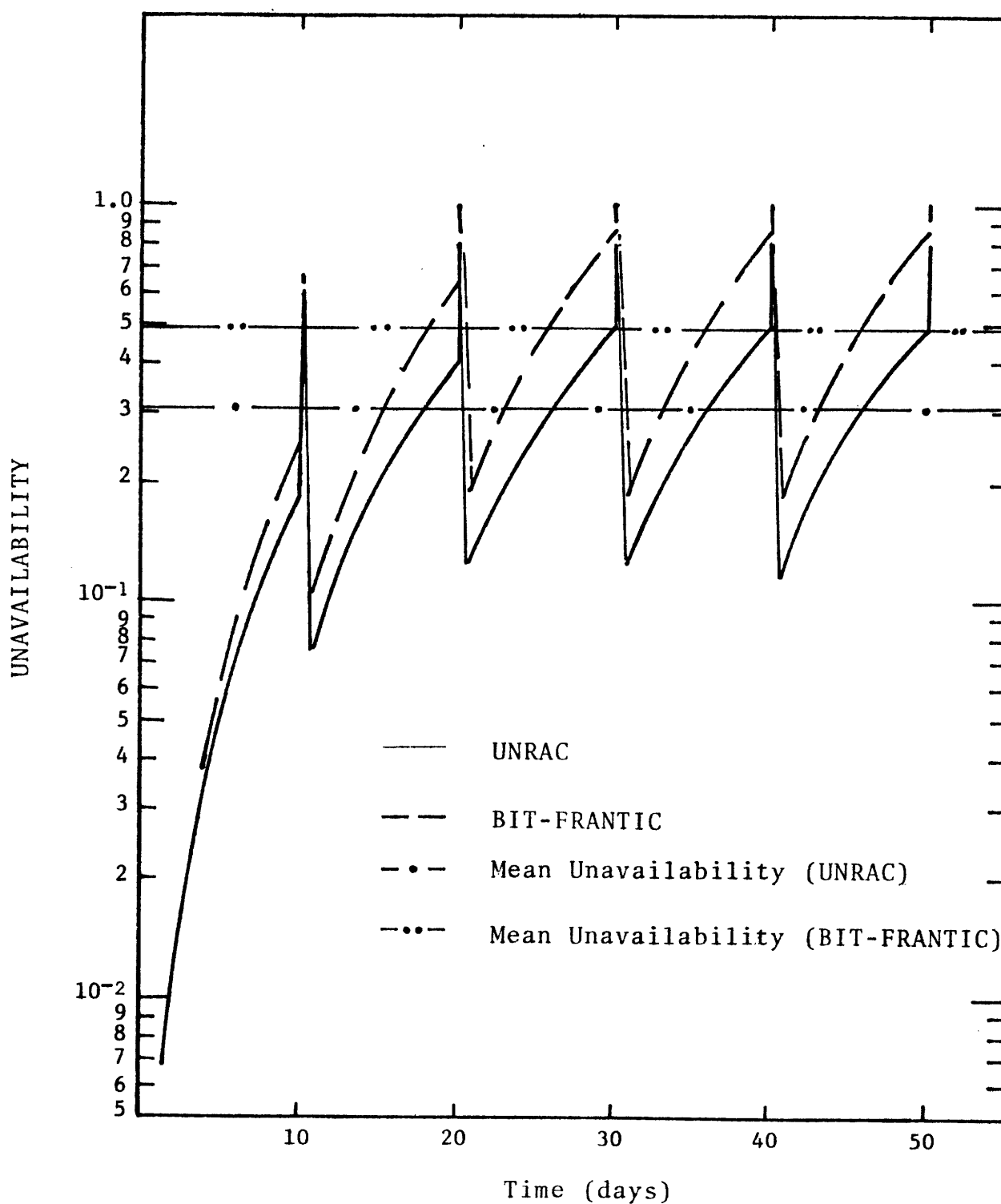


Fig. 4.13: CVCS time dependent unavailabilities as calculated by UNRAC and BIT-FRANTIC

Table 4.12: COMPARISON OF THE CVCS UNAVAILABILITY CALCULATIONS FOR THE EXPONENTIAL AND 2ND ORDER ERLANGIAN REPAIR DISTRIBUTION

<u>Top Event</u>	<u>Repair Distribution</u>	
	<u>Exponential</u>	<u>2nd Order Erlangian</u>
Maximum Unavailability	6.8817E-04	1.7144E-03
Average Unavailability	6.783E-04	1.705E-03
Time to reach to its asymptotic value	Less than 4 days of operation	

uses the second method, the maximum unavailability per demand will be on the order of 1.7×10^{-3} . It is worth mentioning that, the above results are typical numbers and cannot be applied to any CVCS system without validating the data and the model used.

4.5 A High Pressure Injection System (HPIS), Comparison of WASH 1400 and UNRAC

The main purpose of this example is to show the applicability of UNRAC for a large fault tree and the time saving that can be achieved by using the techniques mentioned in Section 3.2.2 for reducing the fault tree prior to quantification. The system chosen for this study is taken from WASH 1400, Appendix II. For the system description the reader is referred to the above reference.

Figure 4.14 depicts the simplified flow diagram of a High Pressure Injection System (HPIS). The original fault tree of the HPIS has more than 8000 minimal cut sets (Garibba et al. (1977)). Since at the present time UNRAC is limited to 3000 minimal cut sets, the original fault tree was further reduced by neglecting the events with negligible unavailability contributions and combining some of the basic events. The resultant tree, shown in Fig. 4.15, consists of a total of 90 basic events and 57 gates. UNRAC generates a total of 1718 minimal cut sets, in which there are 11 with single failure, 104 with double failures and 1603 triple failures per cut sets.

WASH 1400 results for single and double failures are:

$$Q_{\text{singles}} = 1.1 \times 10^{-3}$$

$$Q_{\text{doubles}} = 2.5 \times 10^{-3}$$

The UNRAC results using the data given in WASH 1400 for single and double failures per cut sets are: 1.2×10^{-3} and 2.6×10^{-3} respectively. The results are therefore quite consistent with WASH 1400.

In another run, the failure of delivering water from each pump was considered to be a single basic event. This time the code generates a total of 99 minimal cut sets in which there are 11 with single failure, 79 with double failures and 9 with triple failures per cut sets. The results of this run was exactly the same as the previous run for single and double failures.

The CPU time for the second run was a factor of 8 less than the first run. This reduction in CPU time was the result of combining the basic events input to each of the three OR gates, (FCPA, FCPB, FCPC), shown in Fig. 4.15 (sheet 6) and representing them by one new event. In other words, FCPA, FCPB, and FCPC were considered to be the basic events and Fig. 4.15 (sheet 6) was ignored. This action greatly reduced the number of triple failures and, therefore, results in a smaller number of minimal cut sets.

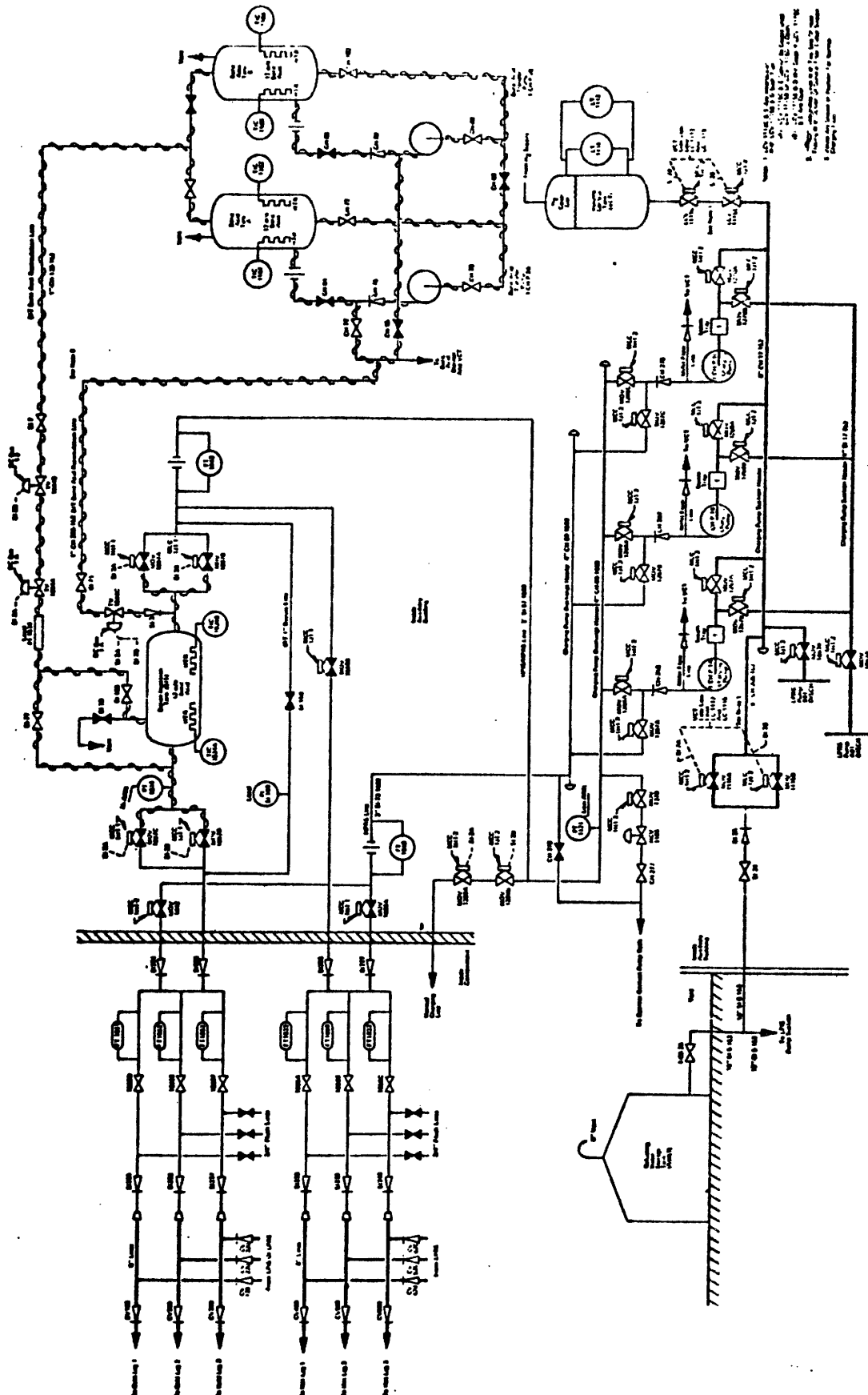


Fig. 4.14: A simplified HPIS diagram (WASH 1400)

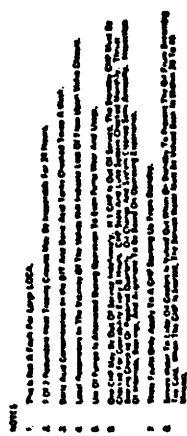


Fig. 4.15: A simplified fault tree diagram of the HPIS (Reduced from WASH 1400)

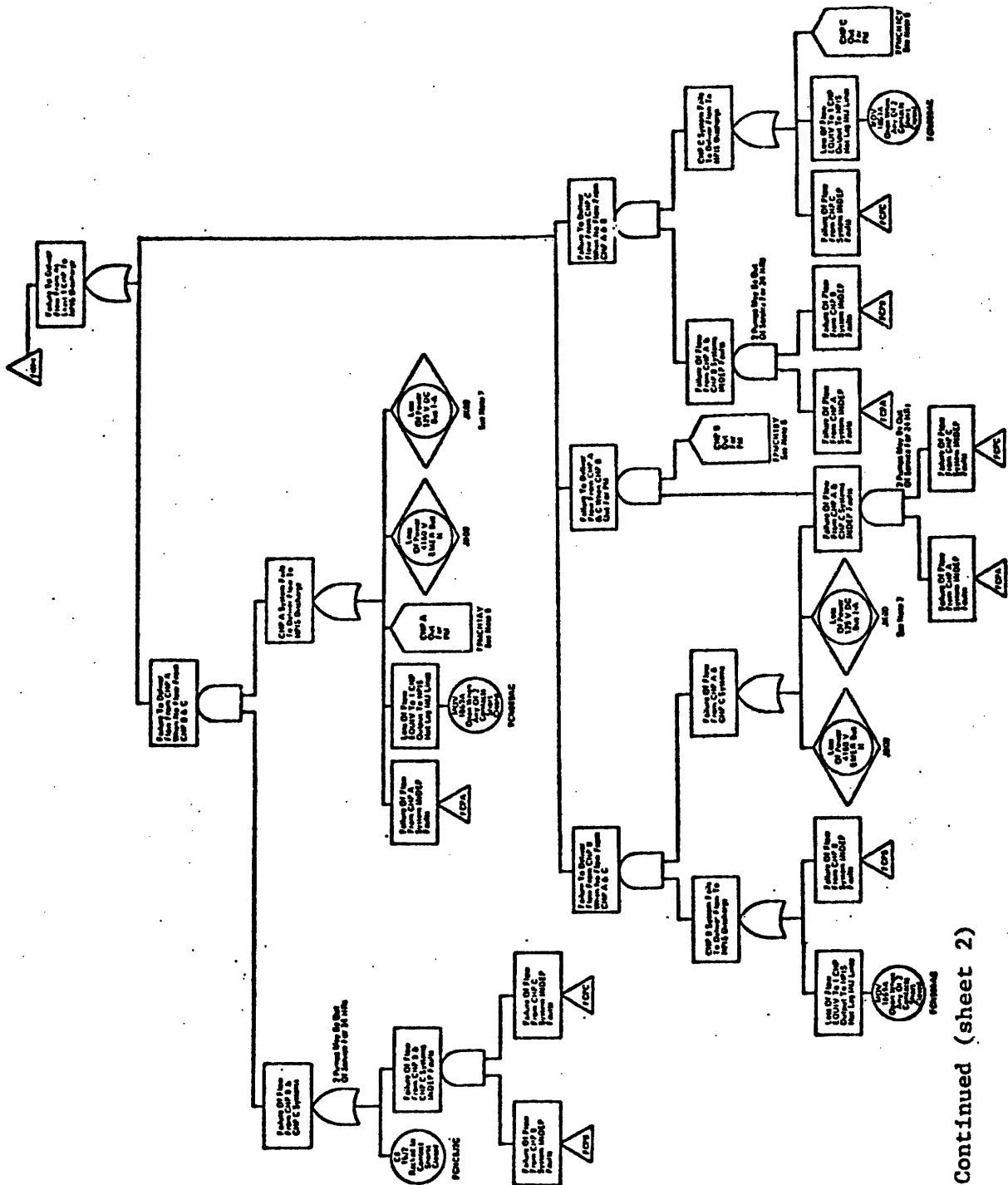


Fig. 4.15 Continued (sheet 2)

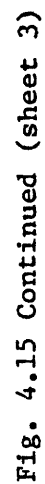


Fig. 4.15 Continued (sheet 3)

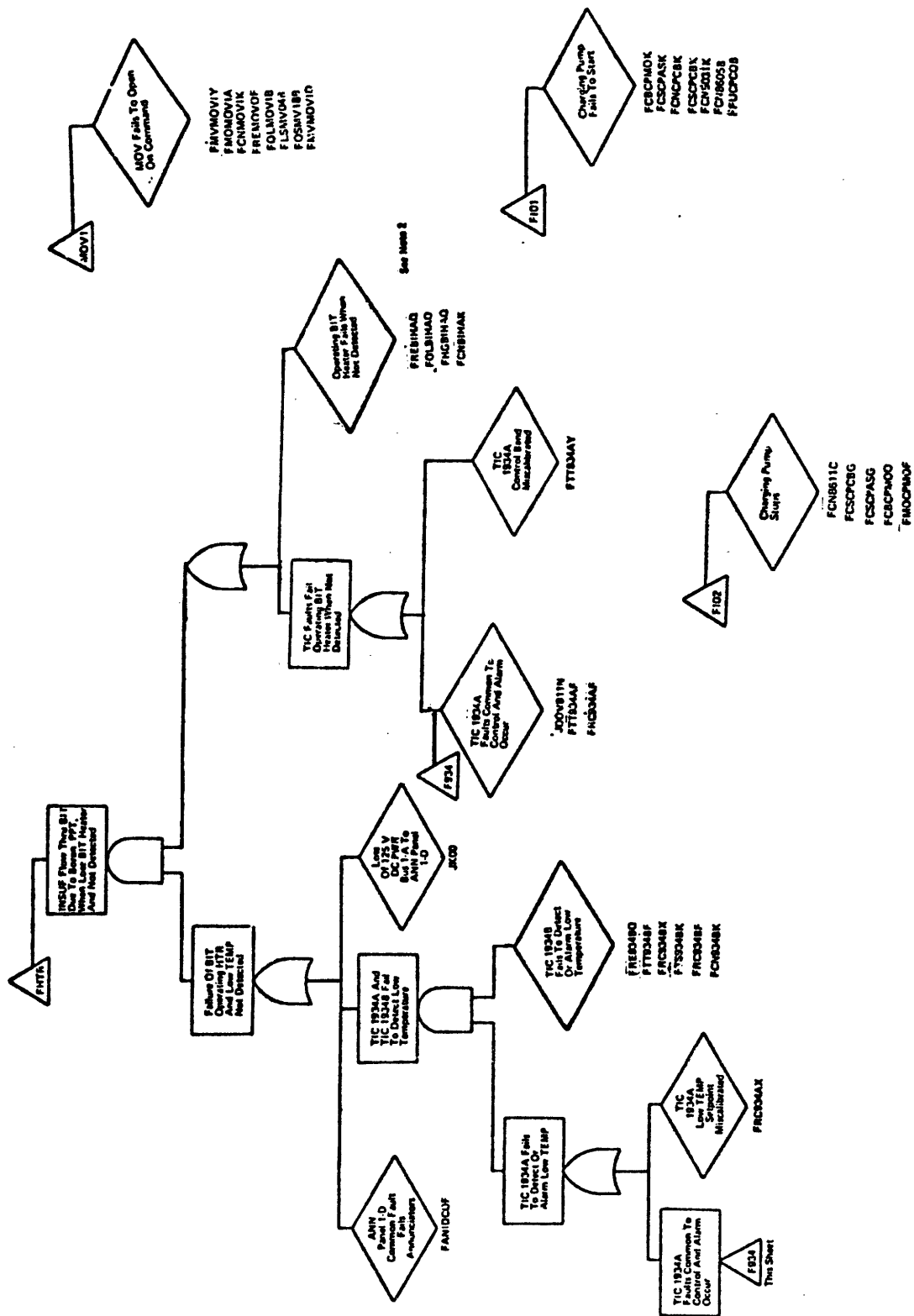


Fig. 4.15 Continued (sheet 4)

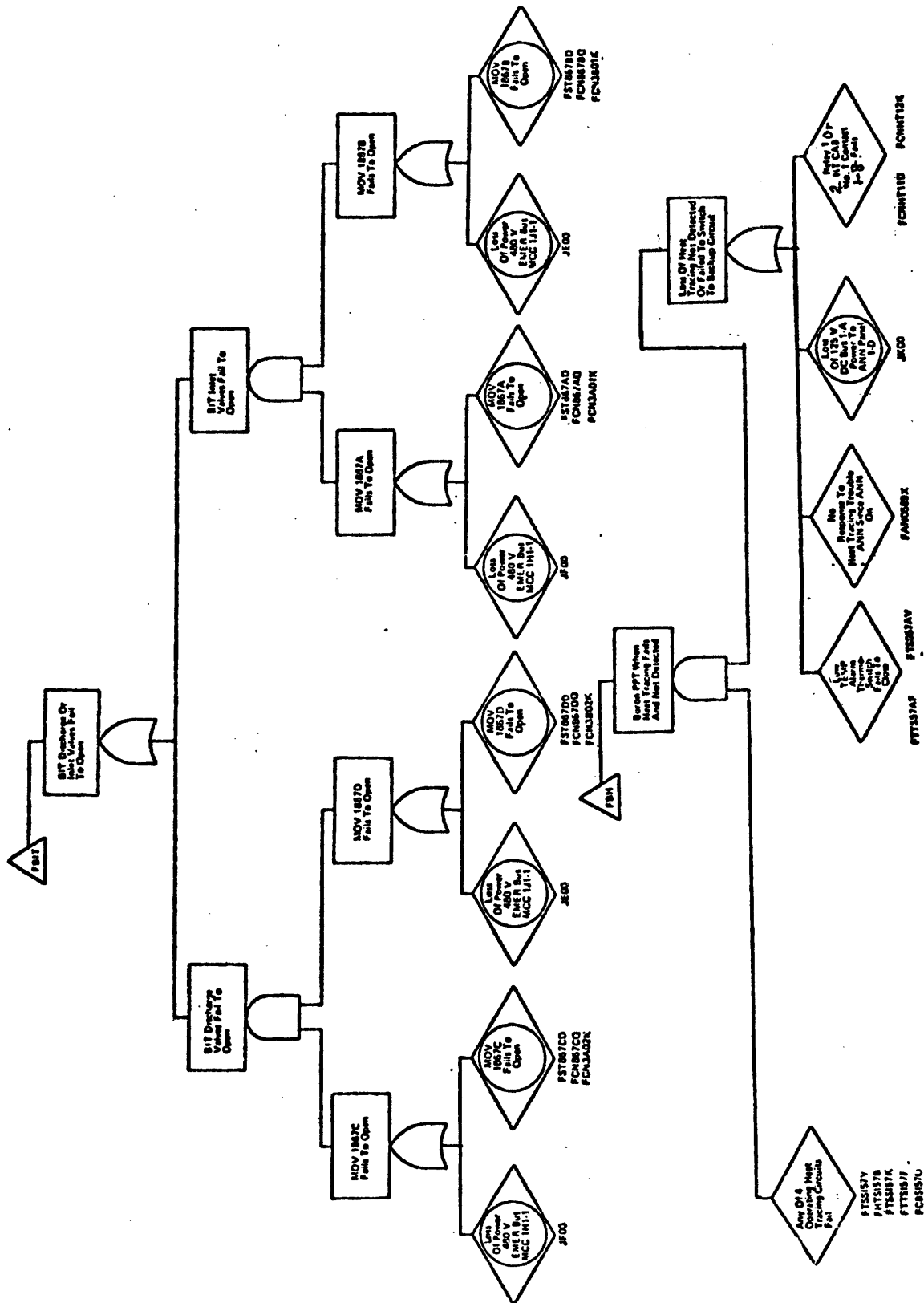


Fig. 4.15 Continued (sheet 5)

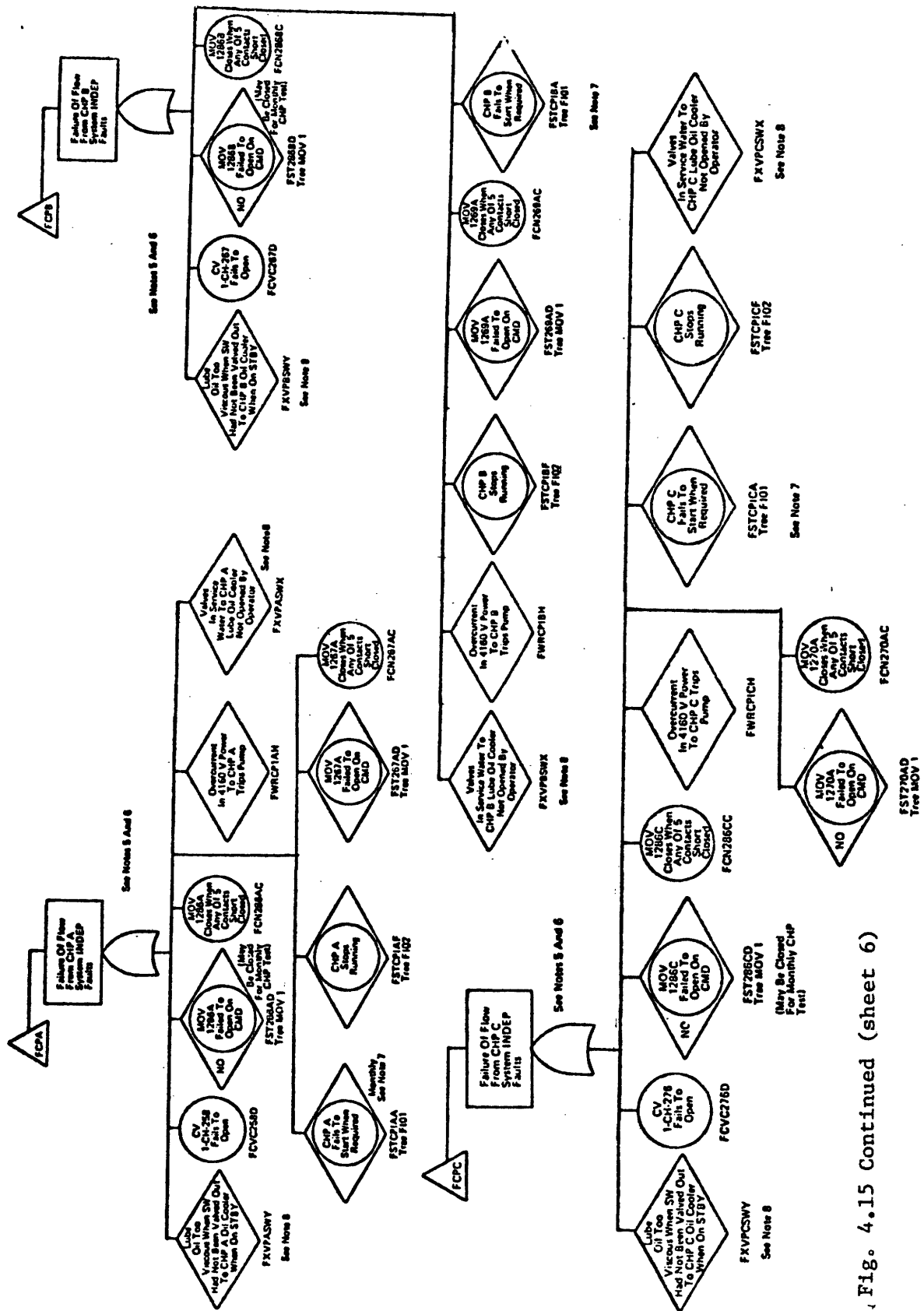
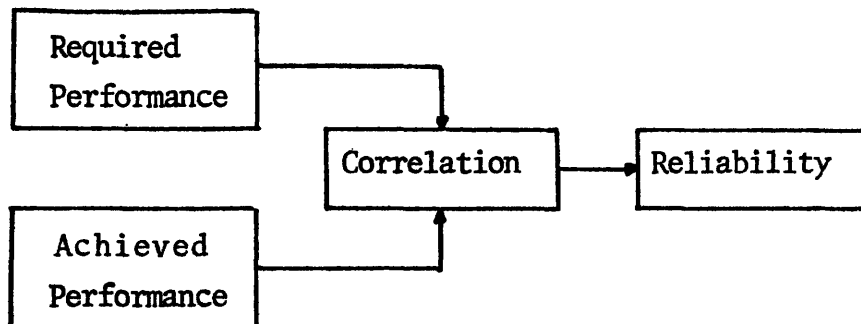


Fig. 4.15 Continued (sheet 6)

CHAPTER 5: SUMMARY, CONCLUSION, AND RECOMMENDATIONS

5.1 Summary and Conclusion

Reliability analysis is a method by which the degree of successful performance of a system under certain stipulated conditions may be expressed in quantitative terms. In order to establish a degree of successful performance, it is necessary to define both the performance requirement of the system and the expected performance achievement of the system. The correlation between these two can then be used to formulate a suitable expression of reliability as illustrated in the following figure, Green and Bourne (1972).



The first step in a reliability analysis, therefore, is to ascertain the pattern of variation for all the relevant performance parameters of the system both from the point of view of requirement as well as from likely achievement. Performance variations may be due not only to the physical attributes of the system and its environment, but also the basic concepts, ideas, and theories which lie behind the system's design. Having established all the appropriate patterns of variations, the system should be rigorously examined to check its ability to work in the required way or fulfill the correct and safe overall function. In order to assess

the appropriate probability expression for each pattern, it is useful to convert the system functional diagram into a logic sequence diagram.

There are two ways to develop the system logic diagram.

- 1) positive logic
- 2) negative logic

The positive logic is called reliability block diagram, whereas the negative logic has acquired different names depending on the types of logic interconnections. They are

- 1) fault tree
- 2) event tree
- 3) cause consequence chart.

Reliability analysis, by using method of fault tree is the most well-known analytic method in use today. Event tree and cause consequence chart (CCC) have also been used; however, by proper rearrangement, one can easily map the CCC into event-tree and fault tree combinations and both can be mapped into a set of fault trees for which the top events are the consequences of the CCC or event tree.

Reliability analysis by using the method of fault tree is known as Fault Tree Analysis (FTA). In FTA, since the system structure logic is composed of a series of negative (failure) logic, the term reliability is always replaced by the term "unreliability" and/or "unavailability". The first step in any FTA is to generate the minimal cut sets which qualitatively delineate the paths of system failure through a simultaneous failure of certain components. The second step in FTA is to find a quantitative value (unreliability or unavailability) for the top event and evaluate the response of the system unavailability to the change in data

and error bounding of the basic components.

In the past decade a number of codes have been written for qualitative and quantitative analysis of the fault tree, and complete analysis requires a combination of several of these codes. Thus, as FTA becomes more widely used it is highly desirable to have a single code which does a complete analysis. The advantage of such a single code is that it reduces the chance of likely error in the input and output process inherent in using multiple codes and it is for this reason that the present study has developed such a code. The objective of this thesis was to develop a code which could generate the minimal cut sets, evaluate the system unavailability (point estimate and/or time dependent), find the quantitative importance of each components involved, and calculate the error bound on the system unavailability evaluation. To develop such a code various aspects of the methodologies used in REBIT[WOLF (1975)], KITT [Vesely (1970)], FRANTIC (Vesely and Goldberg (1977)], SAMPLE [WASH 1400 (1975)], and LIMITS [Lee and Salem (1978)] codes were used as the basis for a single code package.

For generating the minimal cut sets, a portion of REBIT which is called BIT was used. BIT, at the beginning, was limited to 32 components and gates and would only accept AND and OR gates. For this study, it has been modified to accept any kind of gates and its capability has been increased at the present time to 250 components and/or gates. The cut set generator, BIT, was bench marked against MODCUT [Modarres (1979)] and WAMCUT [EPRI (1978)]. It was demonstrated that the BIT code is an efficient and less time consuming code when generating all the minimal cut sets (cf. Section 3.2). To accelerate the code even further a discrimina-

tory procedure based on the cut set size is implemented in the code. This procedure eliminates cut sets larger than a given size.

For the quantification, a general and consistent set of mathematical models for calculating component's unavailabilities was developed and FRANTIC code was restructured to accommodate these models. To evaluate the importance of each component in producing the top event, the methods of Birnbaum and Fussel-Vesely were implemented. It is believed that both methods are necessary in judging the importance of each component.

For error propagation analysis, a routine that allows this simulation (Monte-Carlo) to be carried out on the component's failure characteristics (failure rate, repair rate, test rate, etc.) as well as the average components unavailability was developed. In addition, the routine was written in such a way that one can use different distributions for the component's failure characteristics. To rank the top event unavailabilities in order of their magnitudes to develop a distribution for this parameter a fast procedure similar to the one used in LIMITS code has been employed.

All the above mentioned routines (i.e., cut set generator unavailability evaluator and Monte-Carlo simulator) were coupled and the combined routine is called UNReliability Analysis Code (UNRAC). The UNRAC code was bench marked against BIT-FRANTIC [Karimi and Wolf (1978)], KITT [Vesely (1970)] and PL-MODT [Modarres (1979)] codes. It was demonstrated that UNRAC produces results closer to the KITT results than either BIT-FRANTIC or PL-MODT, (cf. Figure 3.15).

To make UNRAC more versatile, the following features, which are unavailable in any codes in use today, were implemented;

- 1) Using a three state model (cf. Figure 3.9) to represent a normally operating component in order to allow probabilities for revealed faults, non-revealed faults, and false failures to be incorporated in unavailability calculations.
- 2) Incorporating different distributions for the repair time density function in order to analyze the effect of different repair policies on the system unavailability. One such example was done for a periodically tested component and it was found out that the constant repair distribution predicts a higher value for the unavailability per demand than exponential repair distribution and both predict more than a lognormal distribution (cf. Tables 2.1 and 2.2 and Figure 2.9).

Another feature that was incorporated in UNRAC is the capability of finding the importance of the components given that a component or a part of the system has already failed. This is an important measure for the operation management in order to be able to recommend certain procedures for the maintenance of a component or a subsystem by highlighting the important components that should not be perturbed.

The capabilities and models encoded in UNRAC have been demonstrated through a series of examples. Among them was the comparison of UNRAC and WASH 1400 for AFWS. Although WASH 1400 was unable to reveal the strong time dependence of the AFWS unavailability it did report the extreme points unavailabilities which were quite consistent with time dependent analysis. UNRAC estimated the maximum and minimum unavailability to be 1.4×10^{-2} and 5.2×10^{-5} whereas WASH 1400 reported the values of 2.2×10^{-2} and 3.5×10^{-5} respectively (cf. Figures 4.3 and 4.5). It was also found

out that the simple lognormal distribution is a good estimate for the top event distribution (cf. Figures 4.6 and 4.7). This finding supports the validity of WASH 1400 assumption of lognormal for the whole system. Finally from the experience gained during this study we feel that 1200 iterations for Monte-Carlo simulation are adequate for most cases and distributions.

5.2 Recommendations

In the course of this study, it was realized that a minimal cut set generator code requires a large amount of storage capacity. UNRAC requires a total of 800 K byte of storage capacity for all the routines encoded. More than half of this storage is used for generating up to 3000 minimal cut sets with a maximum of 30 components per cut sets. However, this storage requirement could be decreased if a more efficient method of storage and access could be used. One of the methods that might be used is a scratch tape for temporary storage of the variables in the cut set generator subroutine. Another method that might be helpful is the random access storage routine.

The unavailability evaluator in UNRAC is based on the assumption of exponential failure time density function. This distribution cannot accommodate the break-in or wear-out period. Therefore, for better prediction of time dependent unavailabilities distributions like gamma or weibull might be more suitable.

Common-mode/common-cause (cm/cc) failure and its prevention has been a serious concern in the reliability and safety analysis of nuclear systems during the past few years. Several codes have been written to

identify the (cm/cc) candidates and analyze the system quantitatively. Among these, some require a minimal cut set generator to pre-process the fault tree, i.e., COMCAN [Burdick, et al (1976)], BACFIRE (Cate and Fussell(1977)) and some have already coupled with a cut set generator, i.e., COMCAN II [Rasmuson, et al (1978), BACFIRE II (Rooney and Fussell (1978)) and SETS (Worrell and Stack (1977))].

The cut set generator in UNRAC, (BIT), uses bit manipulations for generating and storing the minimal cut sets in the machine word. This feature of UNRAC would make the code very useful for identifying the (cm/cc) candidates. However, before any implementation of the (cm/cc) candidates identifiers a review of the reports by: Rasmuson, et al (1977), Fleming and Raabe (1978) and Edwards and Watson (1979) are highly recommended.

REFERENCES

1. BARLOW R.E. and HUNTER L.C. (1961), "Reliability Analysis of a one-unit System" Operation Res. V.8.No.1 pp 200-208.
2. BARLOW R.E. and LAMBERT H.E. (1975), "Introduction to Fault Tree Analysis" Presented at the conf. on Reliability and Fault Tree Analysis, SIAM 1975, pp 7-35.
3. BARLOW R.E. and PROSCHAN F. (1965), Mathematical Theory of Reliability, John Wiley & Sons.
4. BARLOW R.E. and PROSCHAN F. (1975), Statistical Theory of Reliability and Life Testing, Holt Rinehart and Winston Inc. New York.
5. BATTELLE RICHLAND (1976) "A Preliminary Assessment of Accident Risks in a Conceptual High Level Waste Management System", Battelle Richland, Washington.
6. BENETT R.G. (1975), "On the Analysis of Fault Tree" IEEE Trans. on Rel. Vol. R-24-No.3 (Aug), pp 175-185.
7. BURDICK G.R. et al. (1976) "COMCAN - A Computer Program for Common Cause Analysis" Aerojet Nuclear Company ANCR - 1314 (May)
8. BURDICK G.R. et al. (1977) "Probabilistic Approach to Advanced Reactor Design Optimization" Presented at the Int. Conf. on Nuclear System Reliability Engineering and Risk Assessment held in Gatlinburg TN June 20-24, SIAM
9. CALDAROLA L. (1977) "Unavailability and Failure Intensity of Components" Nucl. Engn'g and Design 44 pp 147-162.

10. CASHWELL E.D. and EVERETT C.J. (1959), A Practical Manual on the Monte-Carlo Method for Random Walk Problems, Pergamon Press
11. CATE C.L. and FUSSELL J.B. (1977) "BAC FIRE - A Computer Code for Common Cause Failure Analysis", University of Tennessee, Knoxville, TN (Aug).
12. CHATTERJEE P. (1974) "Fault Tree Analysis: Min Cut Set Algorithm" ORC 721-2, Operation Research Center, University of California, Berkeley.
13. CONVEYOU R.R. and MACPHERSON R.D. (1967) "Fourier Analysis of Uniform Random Number Generator" Journal of ACM, 14 pp 100-119.
14. COX D.R. (1963) Renewal Theory, John Wiley & Sons, New York.
15. CRAMER, H. (1946) Mathematical Methods of Statistics, Princeton University Press.
16. DAVIS D.J. (1952) "An Analysis of Some Failure Data" Journal of the American Statistical Association, Vol 47 No 258 (June).
17. DIXON W.J. and MASSEY JR., F.J. (1957) Introduction to Statistical Analysis, 2nd Ed. McGraw Hill.
18. ELERATH J.G. and INGRAM G.E. (1979) "Treatment of Dependencies in Reliability Analysis" Proceedings Annual Reliability and Maintainability Symposium, pp 99-103

19. EDWARDS G.T. and WATSON I.A. (1979) "A Study of Common Mode Failures" Report SRD-R-146, U.K. Atomic Energy Establishment, Safety and Reliability Directorate, (July).
20. EPRI-217-2-2 (1975) "Generalized Fault Tree Analysis for Reactor Safety " Electric Power Research Inst. Palo Alto California.
21. EPRI,NP-803 (1978) "WAM-CUT, A Computer Code for Fault Tree Evaluation" Electric Power Research Inst. Palo Alto California.
22. ERDMANN, R.C. et al. (1977) "A Method for Quantifying Logic Models for Safety Analysis" Presented in Int. Conf. on Nuclear System Reliability Engineering and Risk Assessment held in Gatlinburg, TN June 20-24, SIAM.
23. EVANS R.A. (1974) "Fault Tree and Cause-Consequence Chart" IEEE Trans. on Rel. Vol 23-page 1.
24. FELLER W. (1957) An Introduction to Probability Theory and Its Applications, 2nd ed. John Wiley & Sons New York, Vol I.
25. FLEMING K.N. and RAABE P.H. (1978) "A Comparison of Three Methods for the Quantitative Analysis of Common Cause Failure" USDOE Report GA-A-14568, General Atomic Corporation and Development.

26. FUSSELL J.B. (1973) "Fault Tree Analysis - Concepts and Techniques" Proc. NATO Advanced Study Inst. on Generic Techniques of Reliability Assessment, Liverpool (July).
27. FUSSELL J.B. et al. (1974) "MOCUS - A Computer Program to Obtain Minimal Cut Sets from Fault Tree" ANCR-1156, Aerojet Nuclear Company, Idaho Falls, Idaho.
28. FUSSELL J.B. et al. (1974b) "Fault Tree - A State of Art Discussion" IEEE Trans. on Rel. Vol R-23 pp 51-55 (Apr).
29. GARIBBA S. et al. (1977) "Efficient Construction of Minimal Cut Sets from Fault Tree" IEEE Trans. on Rel. Vol R-26 No.2 (June) pp 88-94.
30. GARRICK B.J. (1970) "Principles of Unified Systems Safety Analysis" Nucl. Eng'g and Design Vol 13 pp 245-321.
31. GOKECK O. et al. (1979) "Markov Analysis of Nuclear Plant Failure Dependencies" Proceedings Annual Rel. and Maint. Sympos. pp 104-109.
32. GREEN A.E. and BOURNE A.J. (1972) Reliability Technology, Wiley - Interscience.
33. HAASL D.F. (1965) "Advanced Concepts in Fault Tree Analysis" Presented at System Safety Symposium, June 8 - 9, Seattle Wash.
34. HAHN G.J. and SHAPIRO S.S. (1967) Statistical Models in Engineering, John Wiley & Sons New York.

35. HAMMERSLEY J.M. and HANDSCOMB D.C. (1964) Monte-Carlo Methods, London Methuen & Co. LTD.
36. HENLEY E.J. and KUMAMOTO H. (1978) "Top-Down Algorithm for Obtaining Prime Implicant Sets of Non-Coherent Fault Trees" IEEE Trans. on Rel. Vol R-27-No 4 (Oct) pp 242-249.
37. HENLEY E.J. and WILLIAMS R.A. (1973) Graph Theory In Modern Engineering, Academic Press New York.
38. HOWARD R.A. (1971) Dynamic Probabilistic Systems Vol-I, Vol II John Wiley & Sons.
39. KARIMI R. and WOLF L. (1978) "BIT-FRANTIC - A Convenient and Simple Code Package for Fault Tree Analysis and Unavailability Calculations" Trans Am Nucl Soc, TANSO-30, 1 page 814.
40. KEMENY Y.G. and SNELL J.L. (1960) Finite Markov Chains D. Van Nostrand Co.
41. LAMBERT H.E. (1975) "Measure of Importance of Events and Cut Sets in Fault Tree" Presented in Reliability and Fault Tree Analysis Conf. SIAM Philadelphia pp 77-100.
42. LEE Y.T. and APOSTOLAKIS G.E. (1976) "Probability Intervals for the Top Event Unavailability of Fault Trees" Technical Report, UCLA-ENG-7663, School of Engineering and Applied Science, University of California Los Angeles (June).

43. LEE Y.T. and SALEM S.L. (1977) "Probability Intervals for The Reliability of Complex Systems Using Monte - Carlo Simulation", UCLA-ENG-7758, School of Engineering and Applied Science University of California, Los Angeles, CA.
44. MACLAREN M.D. et al. (1964) "A Fast Procedure for Generating Exponential Random Number" Communication of ACM,7.
45. MANN N.R. et al. (1974) Methods for Statistical Analysis of Reliability and Life Data, John Wiley & Sons Inc.
46. MARSAGLIA G. and BRAY T.A. (1964a) "A Convenient Method for Generating Normal Variables" SIAM Review 6.
47. MARSAGLIA G. et al. (1964b) "Fast Procedure for Generating Normal Random Variables" Communication of ACM,7.
48. MASON, S.J. (1956) "Feedback Theory: Further Properties of Signal Flow Graph" Proc. IRE 44 Page 920.
49. MATTHEWS S.D. (1977) "MOCARS: A Monte-Carlo Simulation Code for Determining the Distribution and Simulation Limits" TREE-1138 (July).
50. MCGRATH E.J. et al. (1975) "Techniques for Efficient Monte-Carlo Simulations" ORNL-RSIC -38 Vol (2).
51. MODARRES M. (1979) "Reliability Analysis of Complex Technical Systems Using The Fault Tree Modularization Technique" Ph.D. Thesis, Nuclear Engineering Dept, Massachusetts Institute of Technology.

52. Nuclear Engineering International (1979) "West German Risk Report Echoes Rasmussen" News Review, Nuclear Engineering International Vol 24 No 290 Page 4 (Sep).
53. PANDE P.K. et al. (1975) "Computerized Fault Tree Analysis, TREEL-MICSUP" Operational Research Center University of California, Berkeley, ORC75-3.
54. PARZEN E. (1960) Modern Probability Theory and Its Applications, John Wiley & Sons Inc. New York.
55. RASMUSON D.M. et al. (1978) "COMCAN II - A Computer Program for Common Cause Failure Analysis" TREE-1289. EGSG Idaho, Inc. (Sep).
56. RASMUSON D.M. et al. (1979) "Common Cause Failure Analysis Techniques; A Review and Comparative Evaluation", Report TREE-1349.
57. RAU J.G. (1970) Optimization and Probability in System Engineering, D. Van Nostrand Reinhold Co. N.Y.
58. RINGOT C. (1978) "French Safety Studies of Pressurized Water Reactor" Nucl. Safety Vol 19 No 4, pp 411-427.
59. ROONEY J.J. and FUSSELL J.B. (1978) "BACKFIRE II - A Computer Program for Common Cause Failure Analysis of Complex Systems" University of Tennessee, Knoxville, TN (Aug).
60. ROSENTAL A. (1975) "A Computer Scientist Looks at Fault Tree Computation" Presented in the Conf. on Reliability and Fault Tree Analysis.

61. SANDLER G.E. (1963) System Reliability Engineering Prentice-Hall, Inc. Englewood Cliff N.J.
62. SEMANDERS S.N. (1971) "ELRAFT, A Computer Program for Efficient Logic Reduction Analysis of Fault Tree" IEEE Trans Nuclear Science Vol NS-18 pp 481-487.
63. SHELL D.L. (1959) "A High-Speed Sorting Procedure" Communication of ACM Vol 12 No 7 pp 30-37.
64. SHOOMAN M.L. (1968) Probabilistic Reliability, An Engineering Approach, McGraw Hill.
65. TAKACS L. (1959) "On a Sojourn Time Problems in the Theory of Stochastics Processes" Trans Amer Math Soc. Vol 93 pp 531-540.
66. TAYLOR J.R. (1974) "Sequential Effects in Failure Mode Analysis" RISO-M-1740 Danish, AEC Roskilde Denmark.
67. VAN SLYKE W.J. and GRIFFING D.E. (1975) "ALLCUTS, A Fast Comprehensive Fault Tree Analysis Code" Atlantic Richfield Hanford Company, Richland Washington, ARH-ST-112 (July).
68. VESELY W.E. (1970) "A Time-Dependent Methodology for Fault Tree Evaluation" Nuclear Eng'g and Design 13 pp 337-360.
69. VESELY W.E. and GOLDBERG F.F. (1977) "FRANTIC - A Computer Code for Time Dependent Unavailability Analysis" NUREG-0193, NRC report, Nuclear Regulatory Commission.

70. VESELY W.E. and NARUM R.R. (1970) "PREP and KITT, Computer Codes for The Automatic Evaluation of a Fault Tree" Idaho Nuclear Corporation, Idaho Falls, Idaho, IN-1349.
71. WASH 1400, Reactor Safety Study, (1975), An Assessment of Accident Risk in U.S. Commercial Nuclear Power Plants" NUREG-75/014 Nuclear Regulatory Commission.
72. WILKES S. (1962) Mathematical Statistics, John Wiley & Sons Inc.
73. WOLF L (1975) "REBIT-A Computer Program for Fault Tree Analysis" Unpublished Work, Department of Nuclear Engineering Massachusetts Institute of Technology.
74. WORREL R.B. (1975) "Using the Set Equation Transformation System (SETS) in Fault Tree Analysis", Reliability and Fault Tree Analysis Conf. SIAM pp 165-185.
75. WORRELL R.B. and STACK D.W. (1977) "Common Cause Analysis Using SETS", SAND77-1832 Sandia Laboratories, Albuquerque New Mexico.

APPENDIX A
DETAILED MATHEMATICAL EXPRESSIONS OF
OF EQNS. (2.39) AND (3.1)

APPENDIX A
DETAILED MATHEMATICAL EXPRESSIONS OF
EQNS. (2.39) AND (3.1)

A.1 Detailed Mathematical Form of the Eqn. (2.39) for Exponential, 2nd order Erlangian and Log-normal Repair Time Density Function

A.1.1 Exponential Repair Time Density Function

If we assume that the repair time density function is:

$$g(t) = \mu \exp(-\mu t) \quad , \text{ with constant } \mu \quad (A.1)$$

then $G(t)$

and the Y term in Eqn. (2.39) will be:

$$G(t) = \int_0^t g(t) = 1 - \exp(-\mu t) \quad (A.2)$$

$$Y = F(T_2 - T_c) - \int_0^{T_2 - T_c} \exp(-\mu \tau) f(t - \tau) d\tau \quad (A.3)$$

And, since we use exponential distribution for the failure density function, i.e., $f(t) = \lambda \exp(-\lambda t)$, Eqn. (A.3) can be reduced to:

$$Y = 1 - \frac{\mu}{\mu - \lambda} [\exp(-\lambda(T_2 - T_c)) \quad (A.4)$$

Substituting for Y from Eqn. (A.3) in Eqn. (2.39) and re-adjusting the reference time to T_c , the resultant equation

will read as:

$$\bar{R}_{\infty}(t - T_c) = 1 - \exp[-\lambda(t - T_c)] + \beta \{ \exp[-\mu(t - T_c)] - \frac{\lambda}{\mu - \lambda} \exp[-\lambda(t - T_c)] - \exp[-\mu(t - T_c)] \} \quad (A.5)$$

$$T_c < t \leq T_2 - T_c$$

where

$$\beta = P_f + (1 - P_f) \frac{1 - \exp[-\lambda(T_2 - T_c)] - \frac{\lambda P_f}{\mu - \lambda} \exp[-\lambda(T_2 - T_c)]}{1 - (1 - P_f) \frac{\lambda}{\mu - \lambda} \exp[-\lambda(T_2 - T_c)]}$$

A.1.2 2nd Order Erlangian Repair Time Density Function

In this case the repair time density function is:

$$g(t) = \mu^2 t \exp(-\mu t) \quad (A.6)$$

and

$$G(t) = 1 - \exp(-\mu t) - \mu t \exp(-\mu t) \quad (A.7)$$

Using Eqn. (A.7), the Y term will be:

$$Y = 1 - \exp[-\lambda(T_2 - T_c)] - \frac{\lambda}{\mu - \lambda} (1 + \frac{\mu}{\mu - \lambda}) \exp[-\lambda(T_2 - T_c)] \quad (A.8)$$

Substituting for Y and G(.) from Eqns. (A.8) and (A.7) in Eqn. (2.39) and readjusting the reference time to T_c the

resultant equation will read:

$$\begin{aligned} \bar{R}_{\infty}(t-T_c) = & 1 - \exp[-\lambda(t-T_c)] + \beta \left\{ \left(1 + \frac{\mu^2}{\mu-\lambda}t\right) \exp(-\mu(t-T_c)) \right. \\ & \left. - \frac{\lambda}{\mu-\lambda} \left(1 + \frac{\mu}{\mu-\lambda}\right) [\exp(-\lambda(t-T_c)) - \exp(-\mu(t-T_c))] \right\} \quad (A.9) \end{aligned}$$

$$T_c < t \leq T_2 - T_c$$

where

$$\beta = P_f + (1-P_f) \frac{1 - \exp(-\lambda(T_2-T_c)) - \frac{\lambda P_f}{\mu-\lambda} \left(1 + \frac{\mu}{\mu-\lambda}\right) \exp(-\lambda(T_2-T_c))}{1 + (1-P_f) \frac{\lambda}{\mu-\lambda} \left(1 + \frac{\mu}{\mu-\lambda}\right) \exp(-\lambda(T_2-T_c))}$$

A.1.3 Log-normal Repair Time Density Function

In this case, $g(t)$ and $G(t)$ are:

$$g(t) = \frac{1}{t\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{\ln t - \mu}{\sigma}\right)^2\right] \quad (A.10)$$

and

$$G(t) = \int_{-\infty}^y \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}x^2\right) dx \quad (A.11)$$

where

$$y = \frac{\ln t - \mu}{\sigma}$$

using $G(\cdot)$, the Y term will read

$$Y = \int_0^{T_2 - T_c} \left(\int_0^{\frac{\ln \pi - \mu}{\sigma}} \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2} x^2) dx \right) f(t - \tau) d\tau \quad (A.12)$$

The above equation cannot be reduced further and it is impossible to find a reduced expression for the Eqn. (2.34). However, Eqn. (2.34) can be solved by successive numerical integration, which has been performed to find the results in the fourth column of Table 2.2.

A.2 Detailed Mathematical Form of the Eqn. (3.1) for Different Repair Density Function

Equation (3.1) was produced by applying the Mason's rules on a general three-state component shown in Fig. 3.9 and is:

$$\phi_{11}^*(s) = \frac{1 - F^*(s)}{s [1 - P_4 F^*(s) P_1 - P_3 F^*(s) P_2 T^*(s) G^*(s)]} \quad (A.13)$$

In this section one of the following repair distributions will be assumed to represent the repair time density function, $g(t)$. They are:

- 1) Exponential
- 2) 2nd order Erlangian
- 3) Log-normal (Approximated by the combinations of exponentials)

We will also assume that the failure and test time distributions to be exponential with the constant parameter λ , ν respectively.

A.2.1 Exponential repair Time Density Function

In this case, $F^*(s)$, $G^*(s)$, and $T^*(s)$ are:

$$F^*(s) = \frac{\lambda}{\lambda + s} \quad (A.14)$$

$$G^*(s) = \frac{\mu}{\mu + s} \quad (A.15)$$

$$T^*(s) = \frac{\nu}{\nu + s} \quad (A.16)$$

Substituting for $F^*(s)$, $G^*(s)$ and $T^*(s)$ from (A.14) to (A.16) in Eqn. (A.13) and simplifying the result we will get:

$$\phi_{11}(s) = \frac{(\mu + s)(\nu + s)}{s[s^2 + as + b]} \quad (\text{A.17})$$

where

$$a = \lambda + \nu + \mu - P_1 P_3 \lambda$$

$$b = \lambda \nu + \nu \mu + P_3 \lambda \mu - P_1 P_3 (\nu + \mu) \lambda$$

To transform Eqn (A.17) into t domain the method of partial fraction will be used. To consider all the possible situations in factorizing the denominator, the results are divided into three distinct parts,

A) The case where $a^2 - 4b > 0$

In this case the denominator can be factorized to:

$$s^2 + as + b = (s + s_1)(s + s_2)$$

where

$$s_1, s_2 = \frac{a}{2} \pm \frac{\sqrt{a^2 - 4b}}{2}$$

and Eqn. (A.17) will become:

$$\phi_{11}(s) = \frac{C_1}{s} + \frac{C_2}{s + s_1} + \frac{C_3}{s + s_2}$$

where

$$C_1 = \frac{\mu_0 \nu}{b}$$

$$C_2 = \frac{(\mu - s_1)(v - s_1)}{-s_1(s_2 - s_1)}$$

$$C_3 = \frac{(\mu - s_2)(v - s_2)}{-s_2(s_1 - s_2)}$$

Using the table of Inverse Laplace transform we will get:

$$\phi_{11}(t) = C_1 + C_2 \exp(-s_1 t) + C_3 \exp(-s_2 t) \quad (\text{A.18})$$

The value of $\phi_{11}(t)$ when t becomes large is equal to C_1 . Therefore:

$$\phi_{11}(\infty) = \frac{\mu \cdot v}{b}$$

B) The case where $a^2 - 4b = 0$

In this case s_1 and s_2 will be equal and by using the routine discussed earlier, $\phi_{11}(t)$ will read:

$$\phi_{11}(t) = C_1 + C_2 \exp(-s_1 t) + C_3 t \exp(-s_1 t) \quad (\text{A.19})$$

where

$$C_1 = \frac{\mu \cdot v}{b}$$

$$C_2 = \frac{\mu + v - 2s_1}{-s_1} + \frac{C_3}{s_1}$$

$$C_3 = \frac{(\mu - s_1)(v - s_1)}{-s_1}$$

$$s_1 = s_2 = \frac{a}{2}$$

C) The case where $a^2 - 4b < 0$ or $4b - a^2 > 0$

If the above situation is true then the Eqn.

(A.18) shall be factorized to:

$$\phi_{11}(s) = \frac{C_1}{s} + \frac{C_2 s + C_3}{[(s + \frac{a}{2})^2 + \frac{4b - a^2}{4}]}$$

Using the table of Inverse Laplace Transform we get

$$\phi_{11}(t) = C_1 + e^{-\frac{a}{2}t} [C_2 \cos kt + \frac{C_3 - C_2 \frac{a}{2}}{k} \sin kt] \quad (A.20)$$

where

$$C_1 = \frac{\mu \cdot v}{b}$$

$$C_2 = 1 - C_1$$

$$C_3 = (\mu + v) - aC_1$$

$$k = \sqrt{\left| \frac{a^2 - 4b}{4} \right|}$$

A.2.2 2nd Order Erlangian Repair Time Density Function

In this case, $g(t)$ has the following form:

$$g(t) = \mu^2 t \exp(-\mu t)$$

and its Laplace transform is:

$$G^*(s) = \frac{\mu^2}{(\mu + s)^2} \quad (A.21)$$

Substituting for $G^*(s)$, $T^*(s)$, and $F^*(s)$ from Eqns. (A.21), (A.16) and (A.14) in Eqn. (A.13) and simplifying the resultant equation we get:

$$\phi_{11}(s) = \frac{(\mu + s)^2 (v + s)}{s[s^3 + as^2 + bs + c]} \quad (\text{A.22})$$

where

$$a = \lambda + v + 2\mu - P_3 P_1 \lambda$$

$$b = s(\lambda + v)\mu + \mu^2 + v\lambda - P_3 P_1 (2\mu + v)$$

$$c = (v + \lambda) \mu^2 + 2\mu v \lambda - P_3 P_1 \lambda (2\mu v - \mu^2) - P_4 \lambda \mu^2$$

To factorize the denominator we have to find its roots. One method is to reduce the denominator to a manageable third order equation. If we substitute $x - \frac{a}{3}$ for s , the denominator will reduce to:

$$\text{Den} = s[x^3 + px + q] \quad (\text{A.23})$$

where

$$p = -\frac{a^3}{3} + b$$

$$q = 2\left(\frac{a}{3}\right)^3 - \frac{ab}{3} + c$$

Then the roots of the bracket term will be

$$[x_1, x_2, x_3] = [D + E, -\frac{D+E}{2} \pm \frac{D-E}{2} \sqrt{-3}]$$

where

$$D = \sqrt[3]{-\frac{q}{2} + \sqrt{Q}}$$

$$E = \sqrt[3]{-\frac{q}{2} - \sqrt{Q}}$$

$$Q = \left(\frac{p}{3}\right)^3 + \left(\frac{q}{2}\right)^2$$

Depending on the value of Q the roots will be changed from the mixture of real and complex to three real roots. Hence we have three distinct situations to evaluate the $\phi_{11}(t)$.

A) The case where $Q > 0$

This case will result in one real root and two complex conjugate roots. By factorizing the Eqn. (A.23) and taking into account that only one root of Eqn. (A.23) should be positive, the Eqn. (A.22) will be factorized to:

$$\phi_{11}(s) = \frac{C_1}{s} + \frac{C_2}{s+y} + \frac{C_3s + C_4}{(s + \frac{Z}{2})^2 + w^2} \quad (\text{A.24})$$

where

$$y = \frac{a}{3} - (D + E)$$

$$\frac{Z}{2} = \frac{a}{3} + \frac{D+E}{2}$$

$$w^2 = \frac{3}{4} \left(\frac{D-E}{2} \right)^2$$

Equating the right sides of Eqns. (A.24) and (A.22) will result in the values of C_1 to C_4

$$C_1 = \frac{\mu^2 v}{e}$$

$$C_2 = \frac{d_2 - y(d_1 - y) - d_3/y}{\left(\frac{Z}{2}\right)^2 + w^2 - y(Z - y)}$$

$$C_3 = 1 - (C_1 + C_2)$$

$$C_4 = d_1 - yC_3 - ZC_2 - (Z + y)C_1$$

$$d_1 = 2\mu + \nu$$

$$d_2 = \mu(\mu + 2\nu)$$

$$d_3 = \mu^2 \nu$$

Knowing the value of C_1 to C_4 , $\phi_{11}(t)$ can be calculated from:

$$\begin{aligned} \phi_{11}(t) = & C_1 + C_2 \exp(-yt) + \exp\left(-\frac{Z}{2}t\right) [C_3 \cos(|w|t) \\ & + \frac{C_4 - C_3 \frac{Z}{2}}{w} \sin(|w|t)] \end{aligned} \quad (A.25)$$

B) The case where $Q = 0$

This case will result in two real distinct roots.

When $Q = 0$, then $D = E$, Eqn. (A.22) can be written as:

$$\phi_{11}(s) = \frac{C_1}{s} + \frac{C_2}{s + \frac{a}{3} - 2D} + \frac{C_3}{\left(s + \frac{a}{3} + D\right)^2} + \frac{C_4}{\left(s + \frac{a}{3} + D\right)}$$

The constants can be calculated from the following equations:

$$C_1 = \frac{\mu^2 \nu}{b}$$

$$C_2 = \frac{(\mu - \frac{a}{3} + 2D)^2 (\mu - \frac{a}{3} + 2D)}{(2D - \frac{a}{3})(3D)^2}$$

$$C_3 = \frac{(\mu - \frac{a}{3} - D)^2 (v - \frac{a}{3} - D)}{(\frac{a}{3} + D)(3D)}$$

$$C_4 = \frac{(\mu - \frac{a}{3} - D)^2 + 2(\mu - \frac{a}{3} - D)(v - \frac{a}{3} - D)}{(\frac{a}{3} + D)(3D)} + \frac{(\frac{a}{3} + 4D)C_3}{(\frac{a}{3} + D)(3D)}$$

Knowing C_1 to C_4 , $\phi_{11}(t)$ can be calculated from

$$\begin{aligned} \phi_{11}(t) = & C_1 + C_2 \exp(-(\frac{a}{3} - 2D)t) + C_3 t \exp(-(\frac{a}{3} + D)t) \\ & + C_4 \exp(-(\frac{a}{3} + D)t) \end{aligned} \quad (A.26)$$

C) The case where $Q < 0$

This case will result in three different real roots.

Therefore, Eqn. (A.22) can be written as:

$$\phi_{11}(s) = \frac{C_1}{s} + \frac{C_2}{s + s_1} + \frac{C_3}{s + s_2} + \frac{C_4}{s + s_3}$$

where

$$s_1 = \frac{a}{3} - 2\sqrt{-\frac{p}{3}} \cos \frac{\theta}{3}$$

$$s_2 = \frac{a}{3} + 2\sqrt{-\frac{p}{3}} \cos(\frac{\theta}{3} + \frac{\pi}{3})$$

$$s_3 = \frac{a}{3} + 2\sqrt{-\frac{p}{3}} \cos(\frac{\theta}{3} - \frac{\pi}{3})$$

$$\theta = \cos^{-1} \left[\frac{-q/2}{\sqrt{-(\frac{p}{3})^3}} \right]$$

$$C_1 = \frac{\mu^2 v}{c}$$

$$C_2 = \frac{(\mu - s_1)^2 (v - s_1)}{-s_1 (s_2 - s_1) (s_3 - s_1)}$$

$$C_3 = \frac{(\mu - s_2)^2 (v - s_2)}{-s_2 (s_1 - s_2) (s_3 - s_2)}$$

$$C_4 = \frac{(\mu - s_3)^2 (v - s_3)}{-s_3 (s_1 - s_2) (s_2 - s_3)}$$

and finally the $\phi_{11}(t)$ will read

$$\begin{aligned} \phi_{11}(t) = & C_1 + C_2 \exp(-s_1 t) + C_3 \exp(-s_3 t) \\ & + C_4 \exp(-s_4 t) \end{aligned} \quad (A.27)$$

A.2.3 Log-normal Repair Time Density Function

Here, we proposed to replace the log-normal distribution with a combination of exponentials. To do so, we approximate the log-normal with the following equation:

$$g(t) = A[\exp(-\lambda_1 t) - \exp(-\lambda_2 t)] \quad (A.28)$$

It was found out that the above equation will result in a good approximation for small tail log-normal, and a conservative value for long tail log-normal distribution. The above results can be seen in Table A.1 to A.3. To find the

Table A.1: COMPARISON OF CUMULATIVE PROBABILITY DISTRIBUTION OF A LOG-NORMAL WITH $\mu = 3.0445$ AND $\sigma = 0.57$ WITH ITS APPROXIMATED COMBINATION OF EXPONENTIALS WITH $\lambda_1 = 6.319 \times 10^{-2}$ AND $\lambda_2 = 1.1263 \times 10^{-1}$

Time Hour	Cumulative Value (%)	
	Log-normal	Approx. Exponential
5	6.0E-01	6.68
10	9.68E	20.3
20	46.6	49.0
30	73.4	70.1
40	87.1	83.2
50	93.6	90.8
60	96.7	95.0
70	98.2	97.3
80	99.0	98.6
90	99.45	99.23
100	99.70	99.60
110	99.81	99.78
120	99.98	99.99
160	99.98	99.99
180	99.99	100.00

Table A.2: COMPARISON OF THE CUMULATIVE
 PROBABILITY DISTRIBUTION OF
 A LOG-NORMAL WITH $\mu = 3.0445$
 AND $\sigma = 0.2$, WITH ITS APPROX-
 IMATED COMBINATION OF EXPO-
 NENTIALS, WITH $\lambda_1 = 8.1783 \times 10^{-2}$
 AND $\lambda_2 = 1.0873 \times 10^{-1}$

Time Hour	Cumulative Value (%)	
	Log-normal	Approx. Exponential
1	0.0	0.417
5	0.00	8.15
10	0.01	24.2
15	4.63	41.1
20	40.4	55.9
25	80.8	67.8
30	96.27	76.93
35	99.46	83.70
40	99.94	88.6
45	99.99	91.8
50	100.00	94.56

Table A.3: COMPARISON OF THE CUMULATIVE PROBABILITY DISTRIBUTION OF A LOG-NORMAL, WITH $\mu = 4.60517$ and $\sigma = 0.66785$, WITH ITS APPROXIMATED COMBINATION OF EXPONENTIALS, WITH $\lambda_1 = 1.0788 \times 10^{-2}$ AND $\lambda_2 = 3.0955 \times 10^{-2}$

Time Hour	Cumulative Value (%)	
	Log-normal	Approx. Exponential
5	0.00	0.3896
10	0.03	1.456
20	0.82	5.1
40	8.5	15.8
60	22.21	28.0
80	36.92	39.74
100	50.00	50.20
120	60.76	59.24
160	75.90	93.10
200	85.01	82.40
300	95.00	93.97
400	98.90	98.93
500	99.20	99.30
600	99.63	99.76
700	99.82	99.92
800	99.91	99.97
900	99.95	99.99

parameters used in Eqn. (A.28), the following properties of the two distributions (log-normal and Eqn. (A.28)) will be equated:

- 1) Mean values
- 2) Median values

To clarify the method, we present the following example. Let us assume that we want to find the approximate distribution of a log-normal (i.e., to find the parameters of Eqn. (A.28)) with the following properties:

$$\mu = 3.0445$$

$$\sigma = 0.57$$

First step is to find the parameter A. The parameter A can be found by normalizing Eqn. (A.28), i.e., $\int_0^{\infty} g(t) dt = 1$, and is:

$$A = \frac{\lambda_1 \lambda_2}{\lambda_2 - \lambda_1} \quad (\text{A.29})$$

The above normalization is necessary because for $g(t)$ to be a probability density function its area over all possible values of t should be equal to one.

The mean value of Eqn. (A.28) is given by:

$$\mu_1' = \int_0^{\infty} t g(t) dt = \frac{\lambda_2 + \lambda_1}{\lambda_1 \lambda_2} \quad (\text{A.30})$$

The median value of the Eqn. (A.28) is a time where the following equation is true:

$$\int_0^{t_m} g(t) = 0.5 \quad (\text{A.31})$$

Now if we replace for μ_1' and t_m from the properties of the log-normal distribution we will have:

$$\begin{aligned} t_m &= e^\mu \\ \mu_1' &= e^{\mu + \frac{1}{2}\sigma^2} \end{aligned} \quad (\text{A.32})$$

By substituting μ_1' and t_m from Eqn. (A.32) in Eqns. (A.30 and (A.31) and simplifying the resultant equations, we will get the following correlation:

$$x \exp\left(-\frac{a}{x}\right) - \exp(-ax) = 0.5(x-1)\exp(a) \quad (\text{A.33})$$

where

$$\begin{aligned} a &= \frac{t_m}{\mu_1'} \\ x &= \frac{\lambda_2}{\lambda_1} \end{aligned}$$

Equation (A.33) can only be solved by numerical iteration. To do so, we simplify the above equation further. We can expand the first exponential term to its third or fifth terms of its series, i.e.:

$$\begin{aligned} \exp\left(-\frac{a}{x}\right) &= 1 - \frac{a}{x} + \frac{1}{2!} \left(\frac{a}{x}\right)^2 - \frac{1}{3!} \left(\frac{a}{x}\right)^3 \\ &\quad + \frac{1}{4!} \left(\frac{a}{x}\right)^4 + \dots \end{aligned} \quad (\text{A.33a})$$

By using the properties of log-normal and the first three terms of the Eqn. (A.33a), Eqn. (A.33) can be put into the following form:

$$x = \frac{1}{0.5e^a - 1} [-e^{-ax} + \frac{a^2}{2x} + 0.5e^a - a] \quad (\text{A.34})$$

where

$$a = 0.85$$

One method of solving such an equation as Eqn. (A.34) is to assume a number for, say x_0 , and find the RMS value of the equation and repeat this action twice. Then find the new adjusted value for x_0 from the following equation (See Appendix A.3):

$$x_0^1 = x_2 + \frac{(x_2 - x_1)^2}{2x_1 - x_0 - x_2} \quad (\text{A.35})$$

If we repeat the above procedure long enough, the value of x_0 will approach the actual root value of the equation. The routine discussed above was used to find the value of x in Eqn. (A.34). It was found out that x has a value of 1.782471 and λ_1 and λ_2 were calculated to be:

$$\lambda_2 = 1.12634 \times 10^{-2}$$

$$\lambda_1 = 6.31899 \times 10^{-2}$$

To check the accuracy of the above results, the cumulative distribution value of $g(t)$ for the median point of

log-normal distribution was calculated from the following equation:

$$\Pr(t \leq e^\mu) = \int_0^{21} g(t) dt = 0.485$$

The above result shows that the matched distribution is a good approximation to the original distribution. However, if the first five terms of the Eqn. (A.33a) were being used (instead of the first three terms), the resultant probability at $t = e^\mu$ would be 0.4928. Here, the difference is less than one percent with the actual probability of the original distribution, which again shows the consistency between the distributions.

Having found the λ_1 and λ_2 parameters, one can easily find the $\phi_{11}(t)$ from the previous repair distribution (2nd order Erlangian). Because the Laplace transform of $g(t)$ is:

$$G^*(s) = \frac{\lambda_1 \lambda_2}{(\lambda_1 + s)(\lambda_2 + s)} \quad (\text{A.36})$$

and can be replaced for Eqn. (A.21) by considering different values for μ . Hence, the parameters a , b and c in the Eqn. (A.22) for matched log-normal will read:

$$\begin{aligned} a &= \lambda + v + \lambda_1 + \lambda_2 - P_3 P_1 \lambda \\ b &= (\lambda + v)(\lambda_1 + \lambda_2) + \lambda_1 \lambda_2 + v\lambda - P_3 P_1 \lambda (\lambda_1 + \lambda_2 + v) \\ c &= (v + \lambda)(\lambda_1 \lambda_2) + (\lambda_1 + \lambda_2)v\lambda - P_3 P_1 (\lambda_1 v + \lambda_2 v + \lambda_1 \lambda_2) \\ &\quad - P_4 \lambda \lambda_1 \lambda_2 \end{aligned} \quad (\text{A.37})$$

And the procedure to find $\phi_{11}(t)$ will be the same as the one which was discussed in the previous section (whereever we have μ^2 and 2μ we will replace them by $\lambda_1\lambda_2$ and $(\lambda_1 + \lambda_2)$ respectively).

A.3 Solution of Equations of the Form $x = f(x)$

In certain cases when the equation can be formed as $x = f(x)$, where f is a known function of x , there is a particularly efficient way of proceeding to the solution.

Let, $x = \xi$ be the root of the above equation; i.e.,

$$\xi = f(\xi) \quad (\text{A.38})$$

Let, x_0 be an initial trial value for the root ξ and define $x_1 = f(x_0)$. And suppose that x_0 is related to exact root by:

$$x_0 = \xi + \epsilon \quad (\text{A.39})$$

Then,

$$x_1 = f(x_0) = f(\xi + \epsilon) = f(\xi) + \epsilon f'(\xi) + O(\epsilon^2)$$

or

$$x_1 = \xi + \epsilon f'(\xi) + O(\epsilon^2) \quad (\text{A.40})$$

Now, let x'_0 denote the next initial value to use, and define it as a linear combination of the first initial trial value x_0 and derived value x_1 .

$$x'_0 = Ax_0 + Bx_1 \quad (\text{A.41})$$

However, by substituting for x_0 and x_1 from Eqns. (A.39) and (A.40) in Eqn. (A.41) we get:

$$\begin{aligned}
 x_0' &= A(\xi + \epsilon) + B(\xi + \epsilon f'(\xi) + O(\epsilon^2)) \\
 x_0' &= (A + B) + \epsilon(A + Bf'(\xi)) + O(\epsilon^2)
 \end{aligned}
 \tag{A.42}$$

Since we wish x_0' to be as close to the root ξ as possible, we set

$$A + B = 1 \tag{A.43}$$

and

$$A + Bf'(\xi) = 0 \tag{A.44}$$

Solving Eqns. (A.43) and (A.44) for A and B:

$$\begin{aligned}
 B &= \frac{1}{1 - f'(\xi)} \\
 A &= 1 - B
 \end{aligned}
 \tag{A.45}$$

The derivative $f'(\xi)$ is not known, however, but it can be approximated by, say, $f'(x_0)$. To avoid differentiating $f(x)$, we can define $x_2 \equiv f(x_1)$, then in turn approximate $f'(x_0)$ by

$$\frac{x_2 - x_1}{x_1 - x_0}.$$

Using this expression together with Eqn. (A.45) in Eqn. (A.41), we will obtain,

$$x_0' = \frac{x_1^2 - x_0 x_2}{2x_1 - x_0 - x_2} = x_2 + \frac{(x_1 - x_2)^2}{2x_1 - x_0 - x_2} \tag{A.46}$$

Thus, x'_0 can be used as a new initial trial value and the process repeated. Convergence is usually very rapid, since the error for each successive trial value is $O(\epsilon^2)$, when the error for the previous trial value is ϵ . However, if $f'(\xi) = 1$, the process if it converges at all, converges more slowly, since from Eqn. (A.44) we cannot then make both $A + B = 1$ and error of order ϵ vanish also.*

* This result is known as Aitken's δ^2 process (See F.B. Hildebrand, Introduction to Numerical Analysis, New York: McGraw Hill C. (1956), p.445).

APPENDIX B

SOME BASIC BACKGROUND AND RELATED
EXAMPLES USED IN SECTION 3.1

2007

2008

2009

2010

2011

2012

2013

2014

2015

2016

2017

2018

2019

2020

2021

2022

2023

APPENDIX B
SOME BASIC BACKGROUND AND RELATED
EXAMPLES USED IN SECTION 3.1

B.1 Mathematical Background for Boolean Operation

Let A, B, C denote sets or collections of sample points in the sample space S. Associated with the event "A occurs" in the event "A does not occur" which is denoted by $\bar{A}$, called complement of event A, i.e., the sample points of S not contained in A. To symbolize "x (any point) is contained in A," we use $x \in A$. However, when we say "the event A is contained in the event B," we use $A \subset B$, and we mean "the occurrence of event A implies the occurrence of event B."

There are two fundamental operations that have had wide application in fault tree analysis; they are union (U) or OR and intersection ($\cap$) or AND operations. The event $A \cup B$ is called the union of A and B, or the event "either A or B (or both) occur." This is equivalent to saying that if $x \in A$ or $x \in B$ (or both), then $x \in A \cup B$, or in closed form we can write:

$$A \cup B = \{x \mid x \in A \text{ or } x \in B\} \quad (\text{B.1})$$

The event $A \cap B$ is called the intersection of A and B, or the event "both A and B occur." And in closed form is

$$A \cap B = \{x \mid x \in A \text{ and } x \in B\}$$

Two sets are disjointed or mutually exclusive if they have no common elements, thus the intersection of two disjoint sets is the null set. This will be written symbolically as

$$A \cap B = \phi$$

The operation between sets or events can be represented using a venn diagram which displays a subset of S as in Fig. B.1. Figure B.1 shows four disjoint sets that form S. And there exist 16 unique subsets which can be produced by all the possible union combinations of the sets 1, 2, 3 and 4. Table B.1 shows the operation between the two sets A and B and summarizes some of the Boolean operation laws. Figure B.1 can be used to check some of the operations mentioned in Table B.1.

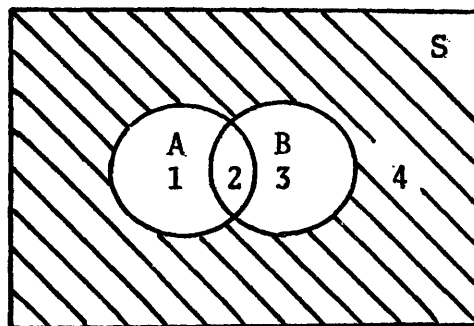


Fig. B.1: A general Venn diagram,

TABLE B.1: LIST OF BOOLEAN OPERATIONS AND IDENTITIES

Property or Operation	Statement of the Operation
1. $\overline{\overline{A}} = A$	The complement of the complement of A is A itself (the Involution Law).
2. $A \cap A = A$ $A \cup A = A$	The intersection/union of A itself is A (the Idempotent Law).
3. $A \cap B = B \cap A$ $A \cup B = B \cup A$	The cumulative law for (intersection/union) of events.
4. $A \cup (A \cap B) = A$ $A \cap (B \cup A) = A$	The absorption law.
5. $\overline{A \cap B} = \overline{A} \cup \overline{B}$ $\overline{A \cup B} = \overline{A} \cap \overline{B}$	The dualization law or (deMorgan's Law).
6. $A \cap \overline{A} = \phi$ $A \cup \overline{A} = S$	The complementary relations. The first states that an event and its complement cannot occur. The second states that an event or its complement must occur.
7. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	The distributive law of intersection with respect to union.
8. $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	The distributive law of union with respect to intersection.
9. $(\overline{A} \cap B) \cup (A \cap \overline{B})$ $(\overline{A} \cap \overline{B}) \cup (A \cap B)$	Exclusive OR, coincidence.

B.2 Application of UNRAC to the Two Fault Tree Examples, for Comparison with MODCUT and WAM-CUT

In this Appendix, the results of the minimal cut sets generated by BIT are given. Figure B.2 shows a fault tree diagram which is used in Modarres (1979). The corresponding minimal cut sets are shown in Table B.2. The component's numbers in Table B.2 are those assigned in the Figure B.2.

Fig. B.3 shows an example fault tree which is used to compare the results of BIT with WAM-CUT (EPRI (1978)). The fault tree is numbered according to the input descriptions to UNRAC. As can be seen, the combination gate is replaced by four gates. Before inputting the tree, gate number 24 which is a NOR gate should be transformed according to Figure 3.3 to AND gate with complemented input events. Table B.3 shows all the minimal cut sets. In comparing to the results given in EPRI(1978), one should note that Table B.3 gives all the minimal cut sets whereas EPRI(1978) reported those cut sets with probability value of 10^{-6} or more. Therefore, in Table B.3, five more minimal cut sets are generated.

Fig. B.2: An example fault tree used to compare the results of BIT and MODCUT.

Table B.2: LIST OF MINIMAL CUT SETS GENERATED
BY BIT FOR FIG. B.2

CUT SET NO. NO. OF COMP. IN C. S. COMPONENTS NOS.

-----	---	---
-----	---	---
1	3	1 2 3
2	3	1 2 4
3	4	1 2 5 16
4	4	1 2 13 17
5	4	1 2 15 19
6	4	1 2 8 20
7	4	1 2 10 21
8	4	1 2 18 23
9	4	1 2 18 25
10	4	1 2 10 26
11	4	1 2 22 28
12	4	1 2 22 24
13	4	1 2 22 23
14	4	1 2 22 29
15	4	1 2 22 30
16	4	1 2 35 36
17	4	1 2 22 27
18	5	1 2 18 33 34
19	5	1 2 22 31 32
20	5	1 2 12 13 35
21	5	1 2 7 8 36
22	5	1 2 9 10 36
23	5	1 2 14 15 35
24	5	1 2 5 6 36
25	5	1 2 11 23 37
26	5	1 2 11 28 36
27	5	1 2 11 24 36
28	5	1 2 11 23 36
29	5	1 2 11 29 36
30	5	1 2 11 30 36
31	5	1 2 23 35 37
32	5	1 2 25 35 37
33	5	1 2 26 35 37
34	5	1 2 11 27 36
35	6	1 2 11 12 13 28
36	6	1 2 11 12 13 24
37	6	1 2 11 12 13 23
38	6	1 2 11 12 13 29
39	6	1 2 11 12 13 30
40	6	1 2 33 34 35 37
41	6	1 2 5 6 23 37
42	6	1 2 5 6 25 37
43	6	1 2 5 6 26 37
44	6	1 2 5 6 12 13
45	6	1 2 7 8 23 37
46	6	1 2 7 8 25 37
47	6	1 2 7 8 26 37
48	6	1 2 9 10 23 37
49	6	1 2 9 10 25 37
50	6	1 2 9 10 26 37
51	6	1 2 11 14 15 28
52	6	1 2 11 14 15 24
53	6	1 2 11 14 15 23
54	6	1 2 11 14 15 29
55	6	1 2 11 14 15 30
56	6	1 2 5 6 14 15

Table B.2 Continued

CUT SET NO.	NO. OF COMP. IN C. S.	COMPONENTS NOS.
57	6	1 2 11 24 26 37
58	6	1 2 7 8 14 15
59	6	1 2 7 8 12 13
60	6	1 2 9 10 14 15
61	6	1 2 11 12 13 27
62	6	1 2 9 10 12 13
63	6	1 2 11 14 15 27
64	6	1 2 11 24 25 37
65	6	1 2 11 25 28 37
66	6	1 2 11 26 28 37
67	6	1 2 11 25 29 37
68	6	1 2 11 25 30 37
69	6	1 2 11 26 29 37
70	6	1 2 11 26 30 37
71	6	1 2 11 31 32 36
72	6	1 2 11 25 27 37
73	6	1 2 11 26 27 37
74	7	1 2 11 26 31 32 37
75	7	1 2 11 29 33 34 37
76	7	1 2 11 30 33 34 37
77	7	1 2 11 28 33 34 37
78	7	1 2 11 27 33 34 37
79	7	1 2 7 8 33 34 37
80	7	1 2 11 24 33 34 37
81	7	1 2 5 6 33 34 37
82	7	1 2 9 10 33 34 37
83	7	1 2 11 12 13 31 32
84	7	1 2 11 25 31 32 37
85	7	1 2 11 14 15 31 32
86	8	1 2 11 31 32 33 34 37

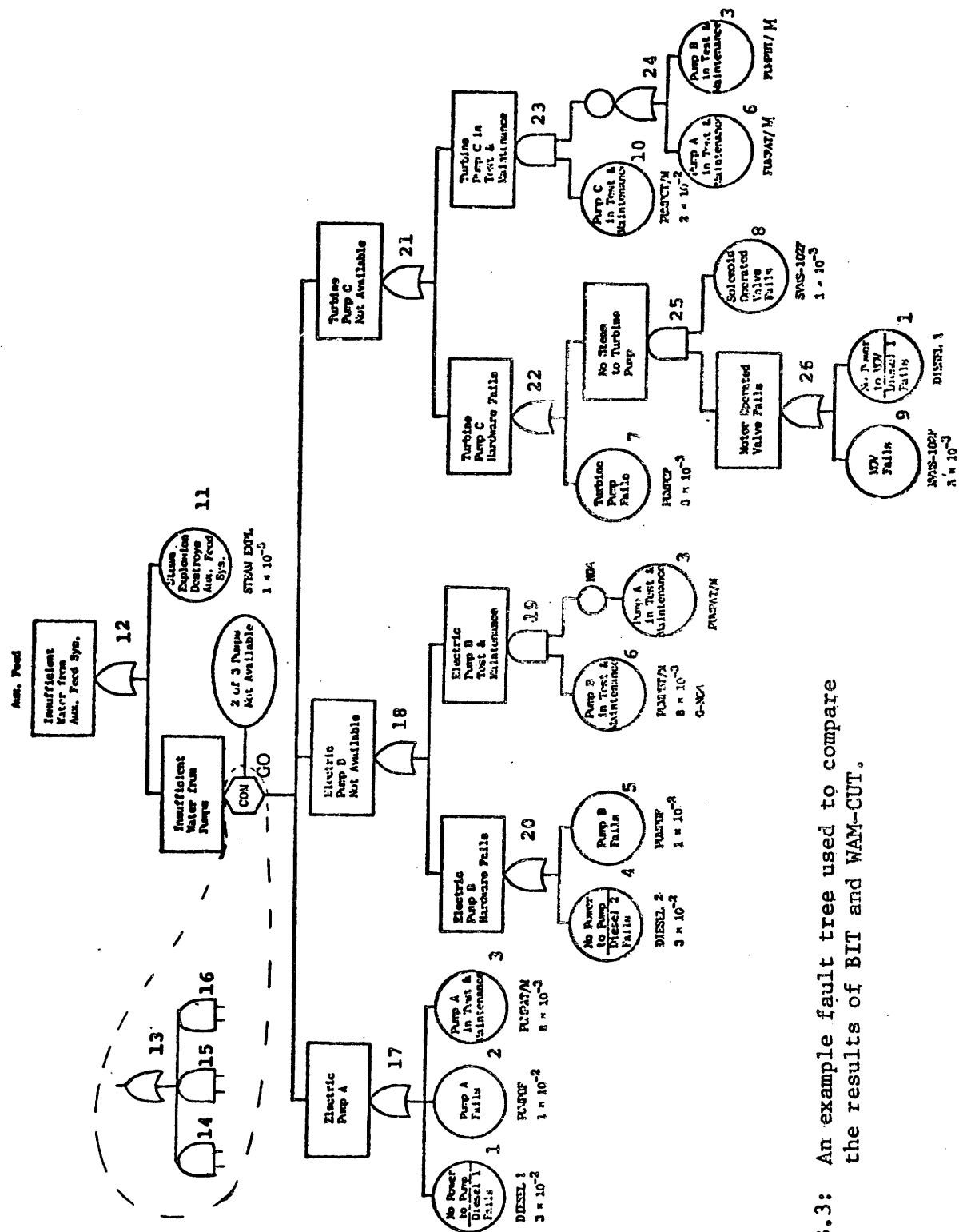


Fig. B.3: An example fault tree used to compare the results of BIT and WAM-CUT.

Table B.3: LIST OF MINIMAL CUT SETS GENERATED
BY BIT FOR FIG. B.3

CUT SET NO.	NO. OF COMP. IN C. S.	COMPONENTS NOS.
1	2	1 4
2	1	11
3	2	1 8
4	3	4 8 9
5	2	2 4
6	2	3 4
7	3	1 -3 6
8	2	1 5
9	3	2 8 9
10	3	3 8 9
11	4	1 -3 -6 10
12	2	1 7
13	4	-3 6 8 9
14	3	5 8 9
15	4	-3 4 -8 10
16	2	4 7
17	2	5 7
18	3	2 -3 6
19	2	2 5
20	4	-3 5 -6 10
21	2	3 5
22	4	2 -3 -6 10
23	2	2 7
24	3	-3 6 7
25	2	3 7

B.3 On the Evaluation of the Equivalent Component's Parameters

In Section 3.2.2, it was mentioned that to decrease CPU time for the cut set generation and improve the efficiency of the code it is advisable to combine some of the non-repetitive components and represent them as a super event or equivalent component when the top event unavailability computation is the desired goal. The above procedure is applicable to components with constant unavailability and those which can be repaired upon the inception of failure (monitored components).

The equivalent component parameters are highly dependent on the type of the gate that the combined components will be input to. Here, we will only consider the OR and AND gate.

The equations that will be summarized in this appendix are taken from a report published by Ross (1975). In that report the asymptotic expected down time and up time of a series and a parallel system of repairable components was analyzed. Table B.4 shows the final equations for equivalent failure rate, λ , and repair rate, μ , of an AND and OR gate. This table was produced by some manipulation of the results given in Ross (1975). In another study, Modarres (1979), an approximated set of equations to evaluate the equivalent λ and μ was reported. For proof the readers will be referred to the above references.

TABLE B.4: EQUIVALENT FAILURE AND REPAIR RATE OF AN AND/OR GATE, FOR MONITORED COMPONENTS AND COMPONENTS WITH CONSTANT UNAVAILABILITY

Gate Type	Equivalent Parameter		Gate Output for Constant Unavailability
	λ	μ	
AND	$\frac{\sum_{i \in x} \mu_i \prod_{i \in x} \frac{\lambda_i}{\mu_i + \lambda_i}}{1 - \prod_{i \in x} \frac{\lambda_i}{\mu_i + \lambda_i}}$	$\sum_{i \in x} \mu_i$	$1 - \prod_{i \in x} (1 - \bar{A}_i)$
OR	$\sum_{i \in x} \lambda_i$	$\frac{[\sum_{i \in x} \lambda_i] \sum_{i \in x} \frac{\mu_i}{\mu_i + \lambda_i}}{1 - \sum_{i \in x} \frac{\mu_i}{\mu_i + \lambda_i}}$	$\prod_{i \in x} \bar{A}_i$

x is a set of n number of components input to the gate.

$\bar{A}_i$ is the constant unavailability of component i .

λ_i, μ_i are the failure and repair rates of the component i .

B.4 References

1. EPRI, NP-803 (1978) "WAM-CUT, A Computer Code for Fault Tree Evaluation" Electric Power Research Inst. Palo Alto California.
2. MODARRES M. (1979) "Reliability Analysis of Complex Technical Systems Using The Fault Tree Modularization Technique" Ph.D. Thesis, Nuclear Engineering Dept., Massachusetts Institute of Technology.
3. ROSS S.M. (1975) "On The Calculation of Asymptotic System Reliability Characteristics", Presented at the Conf. on Reliability and Fault Tree Analysis, SIAM 1975 pp 331 - 350.

PAGES (S) MISSING FROM ORIGINAL

APPENDIX C

ON THE CODE STRUCTURE AND INPUT DESCRIPTION TO UNRAC

C.1 Code Structure

The UNRAC consists of three parts. The first part is the cut set generator which reads in the fault tree description and generates the minimal cut sets. The second part is the unreliability and/or unavailability evaluator which reads in the component's data, i.e., failure rate, test interval, average test time, average repair time, etc., and calculate the top event unavailability and importance of each component involved. Finally, the third part is the Monte-Carlo simulator which reads in the error factor and distribution of each component's failure characteristics, i.e., failure rate, average test time, etc., and simulates the top event distribution to find the error bound on the top event unavailability. Figure C.1 shows the structure of UNRAC discussed above and its related subroutines. The function of each subroutine is described as follows:

- 1-FLOGIC Reads in the fault tree, generates the cut sets, and ranks (optional) the minimal cut sets in order of their size. It calls 3 subroutines; a) MCSN1, b) MCSN2, and c) DECOD. MCSN1 and MCSN2 are used to discard the super sets and duplicated sets and DECOD is used to transform the cut set words into the individual components in each cut set.
- 2-COMDAT Reads in the component's data, checks for any misarrangements in input sequence and prints out the component's input parameters.
- 3-QCAUN Evaluates the average unavailability of each component over a period of one year. It calls two subroutines; a) MONTOR and b) EXAGAM. MONTOR, first sets up proper efficiencies for the monitored components, and then calls SOLN to find the

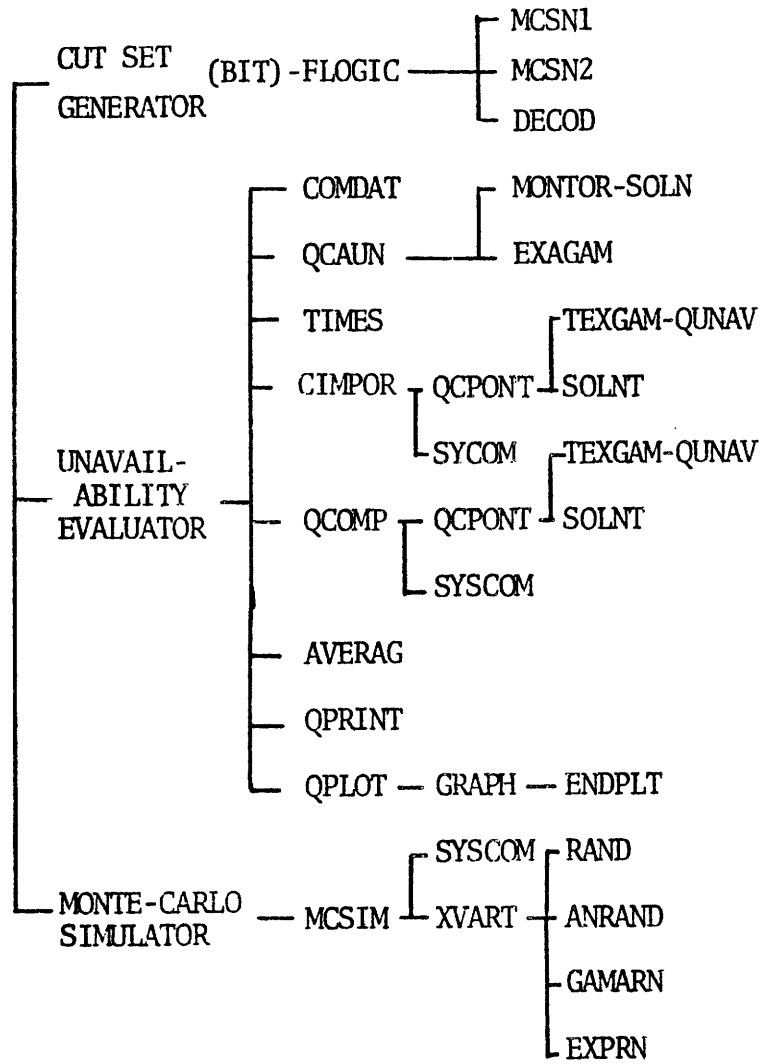


Fig. C.1 : The general routines used in UNRAC

coefficients of the monitored component's unavailability equations and evaluate the average unavailability of each monitored component over a period of one year. EXAGAM is used when the repair distribution of a periodically tested component is exponential and/or special gamma (2nd order erlangian) at the end QCAUN prints the average unavailabilities.

4-TIMES generates all the time points required for the time dependent analysis used in the following routines.

5-CIMPOR Computes the importance of each component based on, either the average values calculated in QCAUN routine or any specified time during the operations it calls QCPONT which will be discussed in the following routine.

6-QCOMP Calculates the top event unavailabilities of the system based on time points generated by TIMES routine. It calls QCPONT and SYSCOM. QCPONT calculates the time dependent unavailability of each component and SYSCOM evaluates the top event unavailability of each component and SYSCOM evaluates the top even unavailability given components unavailabilities. QCPONT calls SOLNT and TEXGAM depending upon the type of component under study. If it is a monitored SOLNT will be called if it is a periodically tested component TEXGAM will be called.

7-AVERAG Finds the average top event unavailability.

8-QPRINT Prints the top events unavailabilities at peaks and detailed time dependent for the requested time intervals.

- 9-QPLOT Plots the top event unavailabilities for the requested type and intervals.
- 10-MCSIM Simulates the top event distributions. It calls XVART and SYSCOM. XVART first simulates each component's failure characteristics according to its distribution by Monte-Carlo sampling. Then finds the each component's unavailability. XVART calls several random number generators, i.e., RAND (uniform), GAMRAN (Gamma), ANRAND (normal) and EXPRN (exponential).

C.2 On the Random Number Generator and Sorting Routine Used in the Monte-Carlo Simulator MCSIM

MCSIM is able to simulate different distributions for the component's failure characteristics. In general, all of the distributions in use today can be simulated by using a combination of uniform, exponential and normal random number generators, see McGrath, et al (1975). For uniform random number generator, the general residual value routine for IBM was employed (cf. Section 3.4.2). To generate the exponential and normal variates the routines of Maclaren, et al (1964) and Marsaglia and Bray (1964) were used. Figures C.2 and C.3 show the flow charts of exponential and normal random number generator.

To rank the top event unavailabilities in order of their magnetude, a sorting routine which was used in LIMITS was incorporated in the MCSIM. Figure C.4 shows the flow chart of the aforementioned routine.

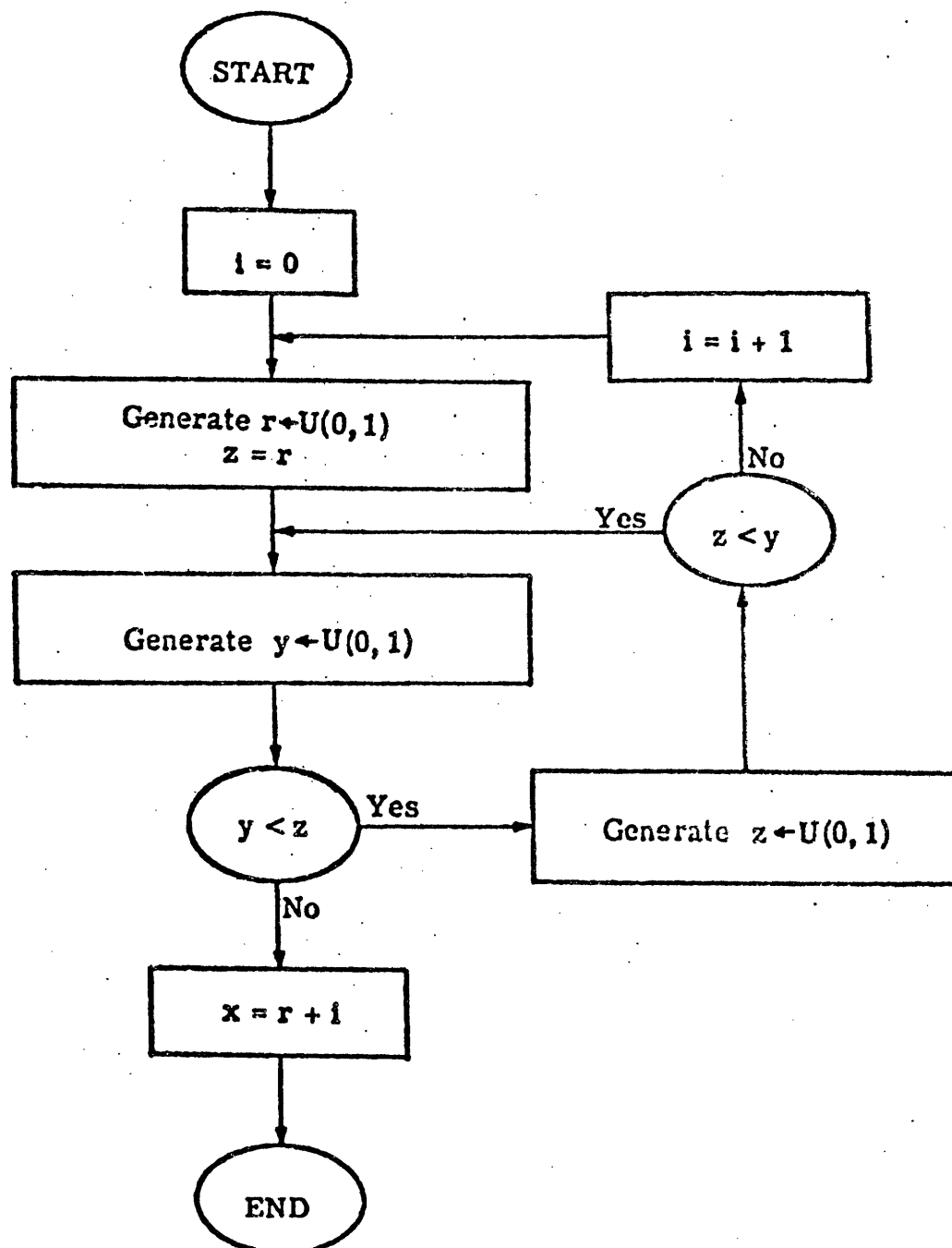


Fig. C.2: Exponential distribution random number generator flow diagram

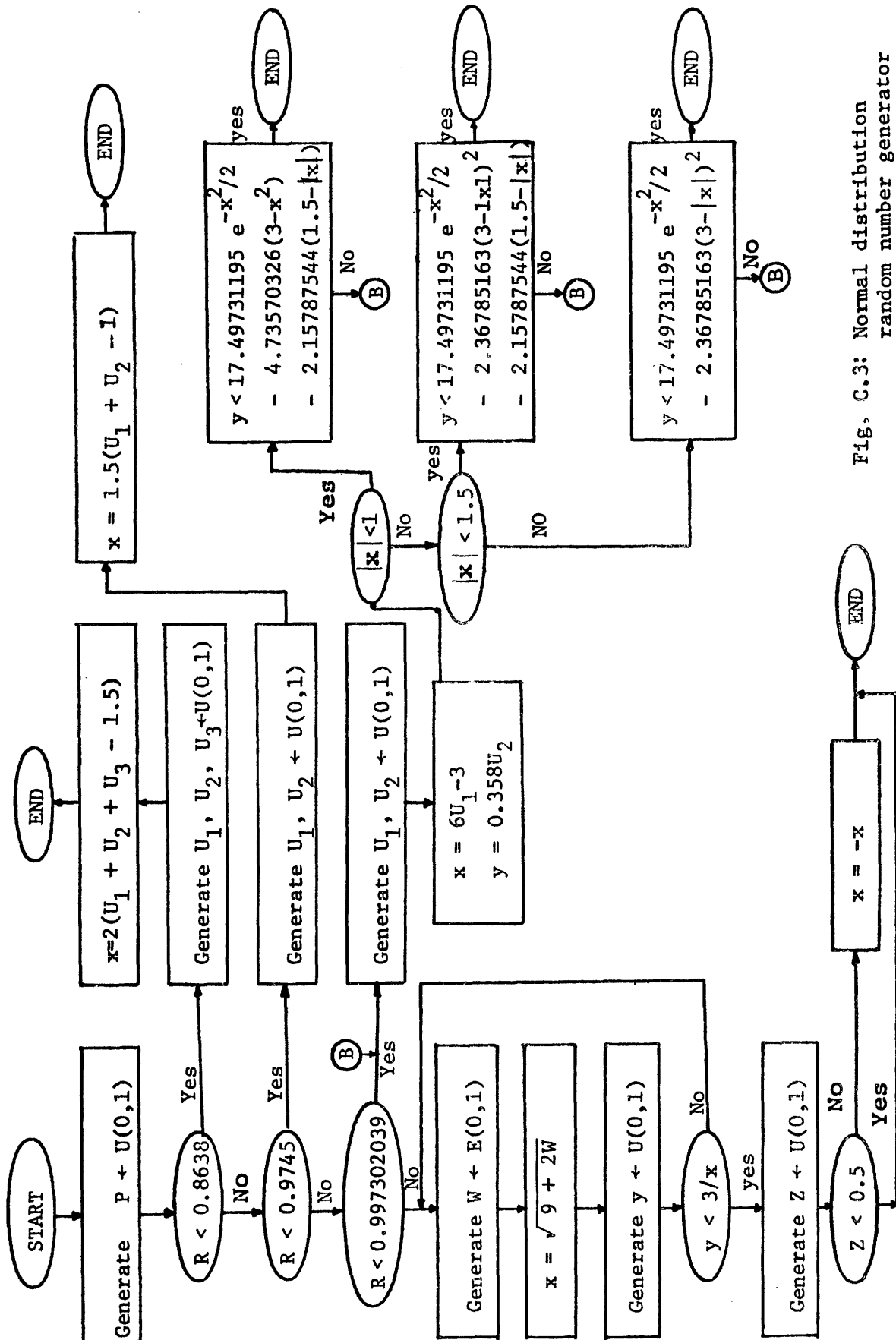
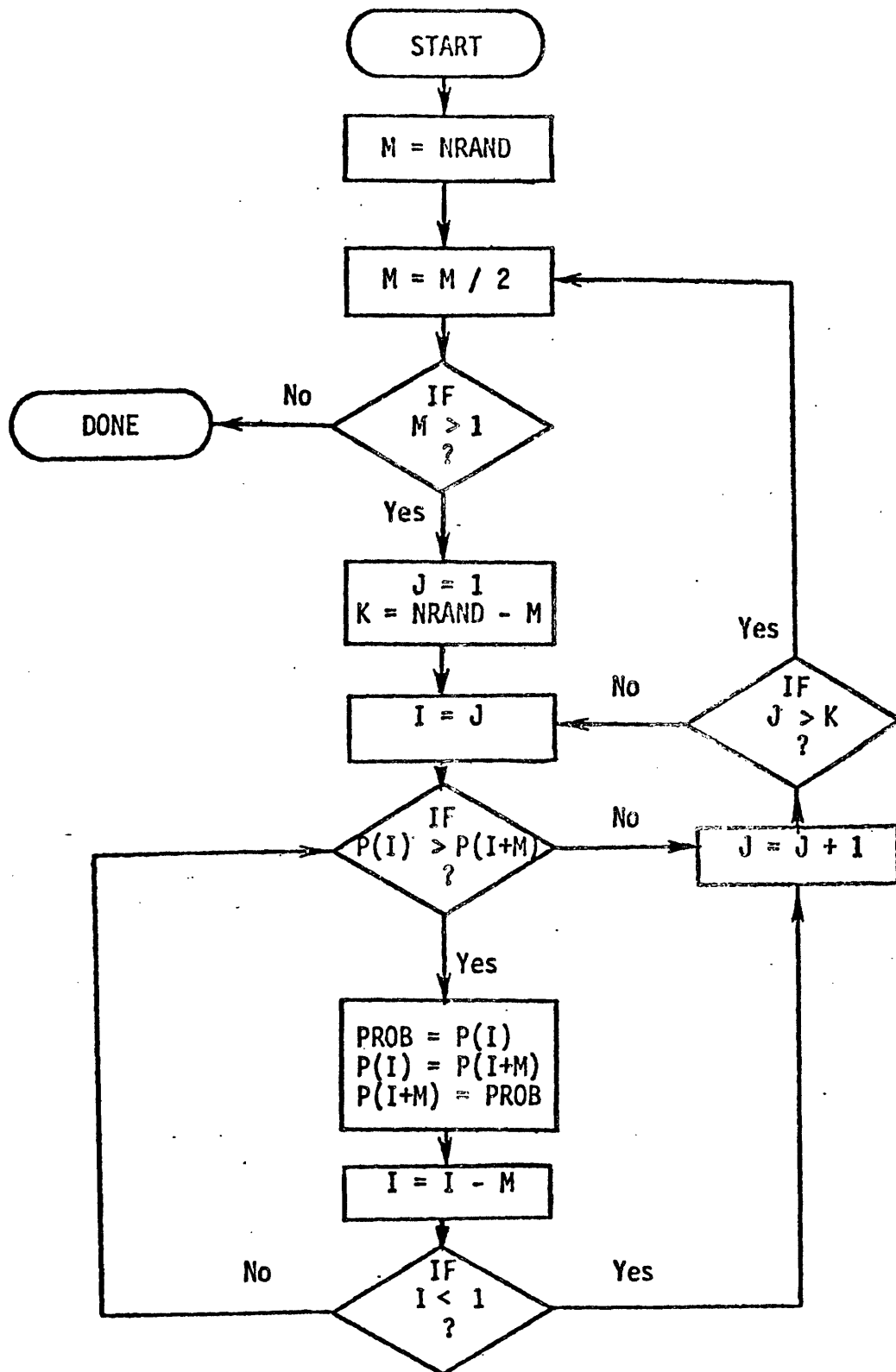


Fig. C.3: Normal distribution random number generator flow diagram



Note: N RAND = Number of random numbers to be sorted
 P (I) = Numbers to be sorted

Fig. C.4: Flow chart of sorting routine used in MCSIM.

C.3 INPUT Description of UNRAC

The UNRAC is written in such a way that it can be used to run one or more cases by inputting the proper sets of data. Each data set of input is called "Data Group". Each data group begins with a keyword card which identifies the data and one or more additional cards containing proper information for that data group. A case consists of 3 data groups:

- 1) Fault tree logic
- 2) Component data.
- 3) Output option.

However, there are a total of 10 data groups needed to run a complete job (demanding all the options encoded). The 10 data groups are described below.

1. Data Group #1 -- TITLE

This data group specifies the title for the case to be run. It consists of two-card types. The first card type is a keyword card containing the characteristics "TITL" in the first 4 columns. The second card type contains 80 characters of text to be used for the output heading. The corresponding variables name and required formats for this data group are:

<u>Card Type #</u>	<u>Columns</u>	<u>Variable Name</u>	<u>Format</u>	<u>Description</u>
1	1 - 4	ANAME	(A 4)	Keyword "TITL"
2	1 - 80	TITLE 1	(20 A 4)	Title for output

2 Data Group #2 -- LOGIC (Fault tree logic)

This data group describes the logic of fault tree to be analyzed. It is identified by a keyword card beginning with the characters "LOGIC". To input the fault tree logic, all the basic events and gates should be numbered in order to minimize the storage requirement. Hence, the following procedures should be employed.

- 1) First assign a number to each basic event (start from 1). For complement events use the negative number of the corresponding events, (i.e., if there exist both A and $\bar{A}$ in a fault tree and the number N has already been assigned to A then -N should be assigned to $\bar{A}$).
- 2) Then assign a number to the top gate, and
- 3) Continue until all the gates are being numbered.

The variable names and the required formats for this group are

<u>Card Type #</u>	<u>Column</u>	<u>Variable Name</u>	<u>Format</u>	<u>Description</u>
1	1 - 4	ANAME	(A 4)	Keyword "LOGI"
2	1 - 3	IMAX	(I ₃ ,	Total number of components in the fault tree
	4 - 6	IMAXT	I ₃ ,	Total number of components plus gates in the fault tree
	7 - 9	LMAX	I ₃	Maximum No. of input to a gate plus two. Presently LMAX Limit is 20.
	10 - 12	NSORT	I ₃ ,	A control card for minimal cut sets printout. If NSORT=1 no need for sorting the min. cut sets. If NSORT=0 order the min. cut sets according to their size.
3	13 - 15	NMAX	I ₃)	Maximum allowable components per cut set.
	1 -	L1(I,J)	(24I3)	Gate Information card(s) Each card starts with the

gate number assigned in the fault tree, gate type (1 for AND 0 for OR) and gate input in any order (components and/or gates).

NOTE: The total number of card type #3 should be equal to IMAXT-IMAX, and the cards should be arranged in a sequential order of the gate numbers starting from the top gate.

3 Data Group #3 -- COMPONENTS

This data group describes the components which make up the system to be evaluated. It is identified by a keyword card beginning with the characters "COMP". This card is followed by an option card containing the characters "NEW" or "UPDATED". The characters "NEW" indicates that the components to be input are to become the effective component set for the case replacing previously input components (if any). "UPDATED" indicates that only the non-blank components parameters are to be used in updating corresponding parameters for previously input components.

After the "NEW" or "UPDATED" card, one card must be entered for each component. This card contains the component number, component name, and 11 parameters describing the reliability data for the component. Zero entries may be left blank.

Under the "NEW" option the component number should be sequential, starting with one. Under the "UPDATED" option, the component numbers are used as keys to identify components to be updated and non-blank fields on the following component cards replace the old values for the corresponding parameters. A negative number must be used to zero out a parameter. The 11 parameters on the component cards are:

<u>Symbol</u>	<u>Name</u>	<u>Description</u>
λ	LAMDA	Failure rate (hr^{-1}) $\times 10^6$
T_2	TEST2	Periodic test interval (days)
T_1	TEST1	First test interval (days)
T_C	TAU	Average test period (hours)
T_R	REPAIR	Average repair time or mean-time-to-repair (hours)
q_0	QAVRD	Test override unavailability
p_f	PTCF	Probability of test - caused failure
θ	INEFF	Detection inefficiency
λ_u	ULAMDA	Undetected failure rate (hr^{-1}) $\times 10^6$
q_d	QRESID	Constant unavailability per demand
	DIST	Repair distribution

If θ is input as a non-zero value, the program will recompute λ as follows: $\lambda_1 = \lambda(1-\theta)$.

If θ is input and λ_u is left blank, the program will compute λ_u as $\lambda_u = \lambda\theta$.

The 11 parameters described above allow the user to specify most types of components under a variety of testing schemes:

- for non-maintained components. The user must provide a non-zero value for λ_u and/or q_d and the rest of the parameters should be left blank.
- for periodically tested components. The user must input a non-zero value for λ and T_2 and optional values for T_1 , T_C , T_R , q_0 , p_f , θ , λ_u , q_d and repair distribution type.
- for monitored components λ and T_R are essential input and T_1 and T_2 should be zero or left blank. If a three state component model is being used then a non-zero value for q_0 and p_f is required. A non-zero value for q_0 refers to the probability that the component does not have any indication for its failure [i.e.,

$p_3=q_0$ in equation (3.1)]. And a non-zero value for p_f indicates the probability that the failure is a spurious one (i.e., $p_1=p_f$ in equation (3.1)).

To specify the repair distribution, the following initials should be used:

- 1) E, for exponential
- 2) u, for uniform and/or constant
- 3) G, for gamma and/or 2nd order erlangian
- 4) M, for lognormal approximated by the combination of exponential.

The data requirement for the case of 1 to 3 are self-explanatory, i.e., T_R is sufficient to express the distribution. For the fourth distribution two parameters are needed [see equation (3.8)]. They are:

$$T_R = 1/\lambda_1$$

$$\lambda_u = \lambda_2$$

Finally, the last card of the components data should contain "-1" in the component numbers field to end the data group. The COMPONENTS data group variable names and their formats are shown below:

<u>Card Type #</u>	<u>Columns</u>	<u>Variable name</u>	<u>Format</u>	<u>Description</u>
1	1- 4	ANAME	(A 4)	Keyword "COMP"
2	1- 4	TYPE	(A 4)	Option 'NEW' or 'UPDAT'
3	1- 5	INDX	(I 5,	Component number
	6-13	NAME	A 8,	Component name
	14-19	LAMDA	F 6.0,	Failure rate x 10^6 /hr.
	20-25	TEST2	F 6.0,	Test interval (days)
	26-31	TEST1	F 6.0,	First test interval (days)
	32-37	TAU	F 6.0,	Average test time (hour)
	38-43	REPAIR	F 6.0,	Average repair time (hour)
	44-49	QOVRD	F 6.0,	Override unavailability
	50-55	PTLF	F 6.0,	Test-caused failure
	56-61	IWEFF	F 6.0,	Detection inefficiency
	62-67	ULAMDA	F 6.0,	Undetected failure rate x 10^6 /hr
	68-77	QRESID	E10.3, IX,	Constant unavailability
	79	DIST	A1)	Repair distribution

NOTE: There should be a total of IMAX (total number of the components) of card type 3. Otherwise the unavailability calculations based on the fault tree input will not be correct. To decrease the time interval between two output results, it is recommended to use a dummy component card. The number for dummy component should be IMAX + 1 and the name should be "DUMMY" punched in columns 6-10. The value for λ should be zero and the value for T_2 should be the desired interval, say 2 days. After the dummy the termination card should be used, i.e., Code "-1" in columns 4 and 5 of the last card.

4 Data Group #4 -- TIME (Optional)

This data group specifies the time period over which component and system unavailability are to be computed. It consists of a keyword card beginning with the characters "TIME" followed by a single card containing the total time (in days) over which the time dependent, instantaneous unavailability is to be computed.

The number of time points generated by the code within the time period is a function of the test intervals, testing times, and repair times of the components. A pair of points is generated where-ever a change in the slope of any component unavailability function occurs.

If the data groups (including the keyword card) is omitted or if a zero is entered for the "time period", the default value of 365 days will be considered by the code. The time data group variable names and their formats are shown below .

<u>Card Type #</u>	<u>Column</u>	<u>Variable Name</u>	<u>Format</u>	<u>Description</u>
1	1- 4	ANAME	(A 4)	Keyword "TIME"
2	1-10	TEND	(F10.0)	Total time period (days)

5 Data Group #5 -- IMPORTANCE (Optional)

This data group is used to request the importance evaluation of a specified set of components or all of the components towards the top event based on either the average unavailability or instantaneous unavailability of the components in the system. The data set is identified by a keyword card beginning with the characters "IMPO". This card is followed by an option card containing the characters "ALL" or "SPECIFIC". "ALL" is used if the calculation of importance of all of the components are desired.

"SPECIFIC" option is used when one or more components of the system have either failed or become inactive in one way or another and it is desired to evaluate the importance of the rest of the components involved.

The third card in this data group should contain the time that the importance calculations must be carried out. This time should be a non-zero value for the "SPECIFIC" case. However, for the "ALL" case the time card can be a blank card. If the blank card is used, the importance calculation will be carried out based on the average unavailability of each component evaluated over a period of one year. For "ALL" case, the time card is the last card in this data set.

In the case of "SPECIFIC" option a total of 5 cards are needed. The fourth card should contain the specified component number which will be assumed to be in failed state or under the test. Finally, the fifth card should contain a list of dependent or inactive components. The IMPORTANCE data group variable names and their formats are shown in the following table:

<u>Card Type #</u>	<u>Columns</u>	<u>Variable Name</u>	<u>Format</u>	<u>Description</u>
1	1- 4	ANAME	(A 4)	Keyword "IMPO"
2	1- 4	TYPE	(A 4)	Optional "ALL" or "SPEC"
3	1-10	TIMPOT	(E10.3)	Time of importance analysis(hr)
IF TYPE = ALL, Then GO TO card data group 6, Otherwise:				
4	1- 3	IMCOMP	(I3,	Specific component's number to be out of service
	4- 6	IDEP	I3)	Total NO. of dependent component to be read in next card.
5	1-	IRELCM(K) K=1, IDEP	(20 I 3)	Component's numbers depend- ent on IMCOMP

6 Data Group #6 -- PRINT (Optional)

This data set is used to request a table printout of the system unavailabilities computed by the program over one or more time intervals (within the input time period) and to specify the number of instantaneous unavailabilities to be separately ranked. The data group is identified by a keyword card beginning with the characters "PRIN". The keyword card must be followed by one card containing the number of time intervals desired and the number of maximum unavailabilities desired. The value input for the number of intervals may be -1, 0, 1, 2, 3, or 4.

If the value input is negative, all system unavailabilities computed are printed, no additional cards are necessary. If the value is zero, any print options previously specified are nullified and the default option (no print) is instituted no addition cards are necessary. If the value is greater than zero, another card containing the end points of the intervals is read, in this case the program will print the system unavailability at all the computed time points that fall within the specified interval(s) including the end points. A maximum of four intervals may be specified.

The maximum unavailability output lists, in decreasing order, the n greatest instantaneous unavailabilities computed by the program. The number of unavailabilities printed (n) has a default value of 12 and may not exceed 100. If the PRINT data set (including the keyword card) is omitted, 12 peaks are printed, and no other system unavailability print-out is produced.

The PRINT data group variables name and the required formats are:

<u>Card Type #</u>	<u>Columns</u>	<u>Variable Name</u>	<u>Format</u>	<u>Description</u>
1	1- 4	ANAME	(A 4)	Keyword "PRIN"
2	1- 5	NPRINT	(I 5,	NO. of time intervals for printing system un- availabilities (-1,0, or 1-4).
	6-10	NPEAK	I 5)	NO. of peaks to be printed, default is 12.
IF NPRINT < 0 GO TO next data group, otherwise:				
3	1-80	(STPRT(I),		Start of Ith interval (Days)
		FINPRT(2),	(8F10.D)	End of Ith interval (Days)
		I=1, 4)		Max. of 4 intervals

7 Data Group #7 -- PLOT (Optional)

This data set is used to specify the time intervals used for plotting the system unavailability. It is identified by a keyword card beginning with characters "PLOT". The keyword card must be followed by a card containing the number of intervals.

If the number of intervals is negative, the plot interval is set to the total time period over which points are computed and no additional cards are necessary. If this value is zero, any plot interval previously input are nullified and the default plot interval is instituted. The default interval is given by;

$$\max_i (T_{1i} + 2 T_{2i} + T_{Ci} + T_{Ri})$$

where T_{1i} , T_{2i} , T_{Ci} and T_{Ri} are the first test interval, second test interval, mean test time and mean repair time of i th components respectively. Thus the default interval is the three largest test cycles of any component, which is often sufficient for establishing the system

behavior.

If the default plot interval exceeds the total time period, then the time period is used instead. If the number of intervals is greater than zero, another card containing the beginning and the end points of each interval is read. A maximum of four intervals may be specified.

Note that unlike the PRINT data set which actually activates the system unavailability printout, the PLOT data set merely sets up the plot intervals which are to be used. Plots must be requested in the RUN (explained in the next data set) data set in order for graphical output to be produced. If plots are requested in the RUN data set, but the PLOT data set (including keyword card) is not input then the default interval described above will be used. The following Table shows the variables name and the required format for PLOT data group.

<u>Card Type #</u>	<u>Columns</u>	<u>Variable Name</u>	<u>Format</u>	<u>Description</u>
1	1- 4	ANAME	(A 4)	Keyword "PLOT"
2	1- 5	NPLOT	(I 5)	NO. of time intervals for plotting system unavailabilities, (-1,0 or 1-4).
IF NPLOT < 0 then GO to next data group; otherwise;				
3	1-80	(STPLT(I),		Start of Ith plot intervals (Days)
		FINPLT(I),		End of Ith plot intervals (Days)
		I=1, 4)	(8F10.0)	Max. of 4 intervals.

8 Data Group #8 -- RUN

This data set initiates the system unavailability calculations. The TITLE, LOGIC, COMPONENTS and optionally the TIME, IMPORTANCE, PRINT, and PLOT parameters must be set up before the RUN data set. The RUN data set is identified by a keyword card beginning with the characters "RUN".

The "RUN" keyword card must be followed by one or more run data cards, where each has the following parameters.

A -- Calculation accuracy number, number code identifying the number of terms to be used in calculating the top event unavailability.

The number coded should be,

"1" if first term approximation (rare event approximation)

"3" if first 3 terms approximation.

"5" if first 5 terms approximation.

is desired. For most evaluations use "1" because "3" and "5" are usually time consuming.

B -- Unavailability option -- four letter code selecting the type of unavailability to be computed where

"FAIL" means compute the instantaneous unavailability based on contributions from component failures only (the between tests contribution).

"TOTL" means compute the instantaneous unavailability based on contributions from failures, testing and repair.

When the unavailability option is left blank, the default value is "TOTL".

C -- PLOT requirement data

C.1 -- x-scale -- four letter code specifying the scaling of the points along the x or time axis where

"NONE" means no plots are produced.

"LIN" means a linear scale is used for the time points.

"MAG" means that the time points are spaced at equal intervals regardless of the actual elapsed time between the points. This produces a plot in which the test and repair contributions are magnified so that the structure of the system unavailability function is easier to see. The indices of the time points are plotted along the x-axis.

"BOTH" means both "LIN" and "MAG" scales are used. Two plots are produced for each y-scale selected.

If x-scale is left blank, the default value is "LIN" when the unavailability option is "FAIL", "BOTH" when the option is "TOTL".

C.2 -- y-scale -- four letter code specifying the scaling of the points along the y or system unavailability axis where

"NONE" means no plots are produced (may be omitted if x-scale = "NONE")

"LIN" means a linear scale is used for the system unavailabilities.

"LOG" means a log scale is used for the system unavailabilities.

"BOTH" means both "LIN" and "LOG" scales are used. Two

plots are produced for each x-scale selected
(e.g., if x-scale = "BOTH" and y-scale = "BOTH",
four plots are produced for each time interval
specified in the "PLOT" data group).

If y-scale is left blank, the default value is "LIN" when
the unavailability option is "FAIL", "LOG" when the un-
availability option is "TOTL".

C.3 -- plot cutoff option -- power of 10 to be used as a lower
bound on system unavailability for plotting (e.g.,
 $-7 = 10^{-7}$). The default is no cutoff.

C.4 -- plot title -- 56 character text to appear as a plot sub-
heading in addition to the title for the case.

A negative system number indicates the end of the RUN
data group. The RUN data group variables are:

<u>Card Type #</u>	<u>Columns</u>	<u>Variable Name</u>	<u>Format</u>	<u>Description</u>
1	1- 4	ANAME	(A 4)	Keyword "RUN"
2	1- 3	NSYS	(I ₃ , 1x	Calculation accuracy No.
	5- 8	QOPT	,A4, 1x	Unavailability option
	10-13	XOPT	,A4, 1x	x-scale
	15-18	XOPT	,A4, 1x	x-scale
	20-23	ICUTO	,I ₄ , 1x	Cutoff option
	25-80	TITLEZ	14 A 4)	Plot title

NOTE: This data group will terminate if a -1 is used for the
calculation accuracy number.

9 Data Group #9 -- SIMULATION (Optional)

This data group is used to request the simulation of the top event distribution using Monte-Carlo technique. The SIMULATION data group is identified by a keyword card beginning with the characters "SIMU". This card must be followed by a card which requests the type of simulation desired. The code can simulate the top event based on:

- 1) error bounds on the average unavailability of each component (WASH 1400 type calculations) and is termed as "type 1 simulation";
- 2) error bounds on the component's failure characteristics, i.e., on failure rate, average test time, average repair time, etc., and is termed as "type 2 simulation".
 --(This routine generates each components parameters distribution and then evaluates the average components unavailabilities by random sampling from the generated parameters distributions.)
- 3) error bounds on the component's failure characteristics, but evaluates the top event distribution at the specific time (s) of operation. (Up to 5 different operational times, and is termed as "type 3 simulation".

The third card type in this data group contains a number for Monte-Carlo iteration, initial random number generator and the degree of accuracy of the evaluation. The uniform random number generator used in this code is the general IBM residual value routine. The general form of the random number generator is:

$$x_{n+1} = a x_n \pmod{2^P}$$

where

p = 31 for IBM 370

a = initial random number generator.

The fourth card type in this data group is for the component's failure characteristics error factors and their respective distributions. Each card carries information about two successive components (i.e., information about comp. #i and i+1), The variables name and required format used in this data group are:

<u>Card Type #</u>	<u>Columns</u>	<u>Variable Name</u>	<u>Format</u>	<u>Description</u>
1	1- 4	ANAME	(A 4)	Keyword "SIMU"
2	1- 3	NJMC	(I3)	A control card to simulate the type of results desired $1 \leq NJMC \leq 7$ NJMC=1, type 1 simulation NJMC=2, type 2 simulation NJMC=3-7, type 3 simulation
3	1- 5	NRAND	(I5,	Number of iteration for Monte-Carlo simulation
	6-20	IX	I15,	Initial random number generator.
	21-26	ACC	F6.0)	Degree of accuracy on the results, default is 0.5.
4	1-80	(FA(I,J),IDTS(I,J), J,1,4), (FA(I+1,J),IDTS(I+1,J), J=1,4) [8(F9.0,A1)]		Error factors on the Component's failure characteristics data and their distributions. The parameters are, failure rate, average test time, average repair time and residual constant unavailability.

IF NJMC < 3 GO TO next data group otherwise;

(cont'd....)

<u>Card Type #</u>	<u>Columns</u>	<u>Variable Name</u>	<u>Format</u>	<u>Description</u>
5	1-50	TMC(I)	(5E10.3)	This card is needed if NJMC $\geq$ 3. This card should contain the time(s) that the simulation should be carried out (time is in hours).

10 Data Group #10 -- END

This card terminates the computations. The card contains the characters 'END' in its first three columns. If more calculations are required simply modify or add the desired parameters using the appropriate data groups and by-pass the END card. It should be remembered that once the fault tree is read in, there is no need to be input again if the same system is being analyzed for all the subsequent cases. To get the time dependent unavailabilities the RUN data group should always follow the changed cases.

When the 'END' card is used the job will terminate and the code will print 'END OF CALCULATION BY UNRAC'. This statement signifies that all the desired evaluations has completed successfully. Otherwise look for misprinted and misplaced card(s).

C.4 References

1. MACLAREN M.D. et al. (1964) "A Fast Procedure for Generating Exponential Random Number" Communication of ACM, 7.
2. MARSAGLIA G. and BRAY T.A. (1964) "A Convenient Method for Generating Normal Variables" SIAM Review 6.

C.5 On The Input Sample Fault Tree

The first step in preparing the input data for UNRAC is to number the fault tree components and/or gates. However, if there exists any NAND, EOR or NOR gates in the fault tree one has to transform them to the basic AND and OR gates according to Fig. 3.3 before numbering the events and gates.

For the sample fault tree, the AFWS is chosen. The fault tree diagram is given in Fig. 4.4. As can be seen, the fault tree is numbered and there exists a total of 23 basic events (components) and 10 gates (ie. $IMAX=23$ and $IMAXT=23+10=33$).

Table C.1 shows the input data which is prepared for the analysis. In this analysis we requested the following information.

- 1) Minimal cut sets of the fault tree to be printed out in any order (ie. $NSORT=1$).
- 2) All the component importance calculations to be carried out based on the average unavailability of each component (ie. $TYPE=ALL$ and $TIMPOT=(Blank)$).
- 3) Average and instantaneous unavailabilities to be calculated based on first term approximation (ie. $NSYS=1$), 12 peaks and complete instantaneous unavailabilities of the system between zero and 60 days to be printed out (ie. $NPEAK=(Blank)$ and $NPRINT=1$).
- 4) Top event simulation to be carried out based on the error factor on the component's failure characteri-

stics (ie. failure rate, average test time, average repair time, etc.) and their assumed log-normal distributions.

Table C.2 gives all the information requested. In sheet one of Table C.2, there exist some useful information about the completeness of the minimal cut set generator, ie., I is the total of minimal cut sets and II is the number of times cut set I is being replaced by II. (for further information see Fig. 3.1). In sheet 5 of Table C.2, a note is printed out to express the values of false failure and unrevealed fault of a monitored component. In sheet 8 of Table C.2, the top event unavailability based on the average components unavailability is given. In sheet 14 of Table C.2, information concerning the error bounds, median value, standard deviation, third and fourth moments and the values of β_1 and β_2 (BETA 1 and BETA 2) are reported. Finally, in sheet 15 the top event cumulative probability distribution with their respective accuracies are given.

NOTE: The CPU time option in the code is inactive, therefore, please disregard the 0 time consumption for the cut set generation.

Table C.1: LIST OF INPUT TO UNRAC FOR THE SAMPLE PROBLEM

```

TITLE
  AUX. FEED WATER RELIABILITY ANALYSIS.
LOGIC
  23 33 8 1 30
  24 0 1 25 28
  25 1 26 27
  26 0 2 3
  27 0 4 5
  28 1 29 30
  29 0 7 6 9 8 10
  30 0 11 12 31
  31 1 32 33
  32 0 13 14 15 16 17 19
  33 0 18 19 20 21 22 23
COMP
NEW
  1 SINS. F 5.100E-07 L
  2 CV 133 1.000E-04 L
  3 CV 131 1.000E-04 L
  4 CV 136 1.000E-04 L
  5 CV 138 1.000E-04 L
  6 CV.TBN 1.000E-04 L
  7 MV.TUR .3 30. 10. 1.5 4.2 1.00 1.000E-03 U
  8 TURPMP 30. 30. 10. 1.5 4.2 1.00 1.000E-03 U
  9 NO STEM 1.000E-03 L
 10 SO 2 F .1 30. 10. 1.5 4.2 1.0 1.000E-03 U
 11 BLD FLD 7.500E-05 L
 12 C.PWR F 1.000E-02 L
 13 CONVP1 5.1 30. 20. 1.5 4.2 1.0 2.000E-03 U
 14 NOPWR2 3.700E-02 L
 15 MVPMP1 .3 30. 20. 1.5 4.2 1.0 1.000E-03 U
 16 CV.EP2 1.000E-04 L
 17 PMP 2 30. 30. 20. 1.5 4.2 1.0 1.000E-03 U
 18 PMP 1 30. 30. 30. 1.5 4.2 1.0 1.000E-03 U
 19 OPR FLT 1.000E-03 L
 20 CV.EP1 1.000E-04 L
 21 MVPMP1 .3 30. 30. 1.5 4.2 1.0 1.000E-03 U
 22 NOPWR1 3.700E-02 L
 23 CONVP1 5.1 30. 30. 1.5 4.2 1.0 2.000E-03 U
 24 DUMMY 5. 5.
-1
TIME
  60.
IMPO
ALL
PRINT
  1
  0.0 60.
RUN
  1 TOTL NONE NONE AUX. FEED WATER SYSTEM REL. ANALYSIS.
-1
SIMUL
  2
 1200 1220703:25 .50

```


Table C.2: RESULTS OF UNRAC FOR THE SAMPLE PROBLEM

TABLE -1 FAULT TREE LOGIC

GATE NO.	GATE TYPE	INPUT COMP. OR GATES							
24	0	1	25	28	0	0	0		
25	1	26	27	0	0	0	0		
26	0	2	3	0	0	0	0		
27	0	4	5	0	0	0	0		
23	1	29	30	0	0	0	0		
29	0	7	6	9	8	10	0		
30	0	11	12	31	0	0	0		
31	1	32	33	0	0	0	0		
32	0	13	14	15	16	17	19		
33	0	18	19	20	21	22	23		

MINO IND LMIN II I
0 0 143 146 145

Table C.2 Continued (sheet 2)

* TOTAL NUMBER OF CUT SET GENERATED = 145 * TIME CONSUMED = .0

TABLE - 2 ;		
CUT SET NO.	NO. OF COMP. IN C. S.	COMPONENTS NOS.
1	1	1
2	2	2 4
3	2	7 11
4	2	3 4
5	2	2 5
6	2	6 11
7	2	9 11
8	2	8 11
9	2	10 11
10	2	7 12
11	3	7 13 18
12	2	3 5
13	2	6 12
14	3	6 13 18
15	2	9 12
16	3	9 13 18
17	2	8 12
18	3	8 13 18
19	2	10 12
20	3	10 13 18
21	3	7 14 18
22	3	7 15 18
23	3	7 16 18
24	3	7 17 18
25	2	7 19
26	3	7 17 23
27	3	7 13 20
28	3	7 13 21
29	3	7 13 22
30	3	7 13 23
31	3	6 14 18
32	3	6 15 18
33	3	6 16 18
34	3	6 17 18
35	2	6 19
36	3	6 17 23
37	3	6 13 20
38	3	6 13 21
39	3	6 13 22
40	3	6 13 23
41	3	9 14 18
42	3	9 15 18
43	3	9 16 18
44	3	9 17 18

Table C.2 Continued (sheet 3)

TABLE - 2 CONTINUED :

CUT SET NO.	NO. OF COMP. IN C. S.	COMPONENTS NOS.
45	2	9 19
46	3	9 17 23
47	3	9 13 20
48	3	9 13 21
49	3	9 13 22
50	3	9 13 23
51	3	8 14 18
52	3	8 15 18
53	3	8 16 18
54	3	8 17 18
55	2	8 19
56	3	8 17 23
57	3	8 13 20
58	3	8 13 21
59	3	8 13 22
60	3	8 13 23
61	3	10 14 18
62	3	10 15 18
63	3	10 16 18
64	3	10 17 18
65	2	10 19
66	3	10 17 23
67	3	10 13 20
68	3	10 13 21
69	3	10 13 22
70	3	10 13 23
71	3	7 17 22
72	3	7 14 20
73	3	7 14 21
74	3	7 14 22
75	3	7 14 23
76	3	7 17 21
77	3	7 15 20
78	3	7 15 21
79	3	7 15 22
80	3	7 15 23
81	3	7 17 20
82	3	7 16 20
83	3	7 16 21
84	3	7 16 22
85	3	7 16 23
86	3	6 17 22
87	3	6 14 20
88	3	6 14 21
89	3	6 14 22

Table C.2 Continued (sheet 4)

TABLE - 2 CONTINUED :

CUT SET NO.	NO. OF COMP. IN C. S.	COMPONENTS NOS.
90	3	6 14 23
91	3	6 17 21
92	3	6 15 20
93	3	6 15 21
94	3	6 15 22
95	3	6 15 23
96	3	6 17 20
97	3	6 16 20
98	3	6 16 21
99	3	6 16 22
100	3	6 16 23
101	3	9 17 22
102	3	9 14 20
103	3	9 14 21
104	3	9 14 22
105	3	9 14 23
106	3	9 17 21
107	3	9 15 20
108	3	9 15 21
109	3	9 15 22
110	3	9 15 23
111	3	9 17 20
112	3	9 16 20
113	3	9 16 21
114	3	9 16 22
115	3	9 16 23
116	3	8 17 22
117	3	8 14 20
118	3	8 14 21
119	3	8 14 22
120	3	8 14 23
121	3	8 17 21
122	3	8 15 20
123	3	8 15 21
124	3	8 15 22
125	3	8 15 23
126	3	8 17 20
127	3	8 16 20
128	3	8 16 21
129	3	8 16 22
130	3	8 16 23
131	3	10 17 22
132	3	10 14 20
133	3	10 14 21
134	3	10 14 22

Table C.2 Continued (sheet 5)

TABLE - 2 CONTINUED :

CUT SET NO.	NO. OF COMP. IN C. S.	COMPONENTS NOS.
135	3	10 14 23
136	3	10 17 21
137	3	10 15 20
138	3	10 15 21
139	3	10 15 22
140	3	10 15 23
141	3	10 17 20
142	3	10 16 20
143	3	10 16 21
144	3	10 16 22
145	3	10 16 23

*** TIME USED TO GENERATE THE MIN. CUT SETS WAS 2.0 SECONDS ***
 *** NOTE : IN THE TABLE 3 COLUMNS 8 , 9 REPRESENT FOUR DIFFERENT VARIABLES;
 WHEN TEST INTERVAL IS ZERO THEN
 COL. 8 IS P3 (PROBAB. THAT MONITOR COMP. NEED THOROUGH TESTING)
 COL. 9 IS P1 (PROB. THAT THE FAIL. IS SPURIOUS)
 WHEN TEST INTERVAL IS NON-ZERO THE VALUES ARE AS WRITTEN.

AUX. FEED WATER RELIABILITY ANALYSIS.

TABLE - 3 ;

***** INPUT COMPONENT PARAMETERS *****

COMP NUMB	COMP NAME	FAILURE RATE (PER HR)	TEST INTERVAL (DAYS)	FIRST INTERVAL (DAYS)	TIME FOR TESTING (HRS)	TIME FOR REPAIR (HRS)	OVRRIDE UNAVAIL	PROB OF TEST-CAUSED FAILURE	TEST INEFF	UNDETECT FAIL RATE (PER HR)	RESIDUAL UNAVAIL	REPAIR DIST.
1	SINS. F	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.10E-07	L
2	CV 133	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04	L
3	CV 131	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04	L
4	CV 136	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04	L
5	CV 139	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04	L
6	CV.TBN	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04	L
7	MV.TUR	3.00E-07	3.00E+01	1.00E+01	1.50E+00	4.20E+00	1.00E+00	0.0	0.0	0.0	1.00E-03	U
8	TURPMP	3.00E-05	3.00E+01	1.00E+01	1.50E+00	4.20E+00	1.00E+00	0.0	0.0	0.0	1.00E-03	U
9	NO STEM	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-03	L
10	SO 2 F	1.00E-07	3.00E+01	1.00E+01	1.50E+00	4.20E+00	1.00E+00	0.0	0.0	0.0	1.00E-03	U
11	BLD FLD	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.50E-05	L
12	C.PWR F	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-02	L
13	CONVP1	5.10E-06	3.00E+01	2.00E+01	1.50E+00	4.20E+00	1.00E+00	0.0	0.0	0.0	2.00E-03	U
14	NOPWR2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.70E-02	L
15	MVPMPI	3.00E-07	3.00E+01	2.00E+01	1.50E+00	4.20E+00	1.00E+00	0.0	0.0	0.0	1.00E-03	U
16	CV.EP2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04	L
17	PMP 2	3.00E-05	3.00E+01	2.00E+01	1.50E+00	4.20E+00	1.00E+00	0.0	0.0	0.0	1.00E-03	U
18	PMP 1	3.00E-05	3.00E+01	3.00E+01	1.50E+00	4.20E+00	1.00E+00	0.0	0.0	0.0	1.00E-03	U
19	OPER FLY	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-03	L
20	CV.EP1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.00E-04	L
21	MVPMPI	3.00E-07	3.00E+01	3.00E+01	1.50E+00	4.20E+00	1.00E+00	0.0	0.0	0.0	1.00E-03	U
22	NOPWR1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.70E-02	L
23	CONVP1	5.10E-06	3.00E+01	3.00E+01	1.50E+00	4.20E+00	1.00E+00	0.0	0.0	0.0	2.00E-03	U
24	DUMNY	0.0	5.00E+00	5.00E+00	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

Table C.2 Continued (sheet 6)

AUX. FEED WATER RELIABILITY ANALYSIS.

TABLE - 4 :

 ***** COMPONENT MEAN UNAVAILABILITIES *****

COMP NUMB	COMP NAME	TOTAL UNAVAIL	UNAVAIL DUE TO FAILURES	% OF TOTAL	UNAVAIL DUE TO TESTING	% OF TOTAL	UNAVAIL DUE TO REPAIRS	% OF TOTAL	UNAVAIL DUE TO UNDET FAIL.	% OF TOTAL	RESIDUAL UNAVAIL	% OF TOTAL
1	SINS. F	5.100E-07	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	5.100E-07	100.00
2	CV 133	1.000E-04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000E-04	100.00
3	CV 131	1.000E-04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000E-04	100.00
4	CV 136	1.000E-04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000E-04	100.00
5	CV 138	1.000E-04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000E-04	100.00
6	CV.TBN	1.000E-04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000E-04	100.00
7	MV.TUR	3.303E-03	1.054E-04	3.19	2.204E-03	66.75	1.260E-06	0.04	0.0	0.0	9.916E-04	30.03
8	TURPMP	1.378E-02	1.046E-02	75.91	2.204E-03	15.99	1.247E-04	0.90	0.0	0.0	9.916E-04	7.19
9	NO STEM	1.000E-03	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000E-03	100.00
10	SD 2 F	3.232E-03	3.512E-05	1.09	2.204E-03	68.21	4.202E-07	0.01	0.0	0.0	9.916E-04	30.69
11	3LD FLD	7.500E-05	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	7.500E-05	100.00
12	C.PWR F	1.000E-02	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000E-02	100.00
13	CONVP1	5.939E-03	1.793E-03	30.18	2.141E-03	36.06	2.143E-05	0.36	0.0	0.0	1.984E-02	33.40
14	NDPWR2	3.700E-02	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.700E-02	100.00
15	MVPMP1	3.240E-03	1.054E-04	3.25	2.141E-03	66.09	1.260E-06	0.04	0.0	0.0	9.919E-04	30.61
16	CV.EP2	1.000E-04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000E-04	100.00
17	PMP 2	1.373E-02	1.047E-02	76.27	2.141E-03	15.60	1.247E-04	0.91	0.0	0.0	9.919E-04	7.22
18	PMP 1	1.387E-02	1.068E-02	76.95	2.082E-03	15.01	1.247E-04	0.90	0.0	0.0	9.921E-04	7.15
19	OPER FLT	1.000E-03	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000E-03	100.00
20	CV.EP1	1.000E-04	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000E-04	100.00
21	MVPMP1	3.183E-03	1.075E-04	3.38	2.082E-03	65.41	1.260E-06	0.04	0.0	0.0	9.921E-04	31.17
22	NDPWR1	3.700E-02	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.700E-02	100.00
23	CONVP1	5.915E-03	1.828E-03	30.90	2.082E-03	35.20	2.143E-05	0.36	0.0	0.0	1.984E-03	33.54
24	DUMMY	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table C.2 Continued (sheet 7)

Table C.2 Continued (sheet 8)

INPUT DATA FOR IMPORTANCE ANALYSIS.
TYPE OF CALCULATION = ALL
TIME OF EVALUATION = 0.0
AVERAGE OR POINT ESTIMATE TOP EVENT UNRELIABILITY IS= 3.1497D-04

Table C.2 Continued (sheet 9)

TABLE - 5 ;

FUSSELL-VESELY AND BIRNBAUM'S MEASURE OF IMPORTANCE

COMPONENT NO.	FUSSELL-VESELY MEASURE	BIRNBAUM'S MEASURE
1	1.6192E-03	1.0000E+00
2	6.3498E-05	2.0000E-04
3	6.3498E-05	2.0000E-04
4	6.3498E-05	2.0000E-04
5	6.3498E-05	2.0000E-04
6	4.6607E-03	1.4668E-02
7	1.5392E-01	1.4680E-02
8	6.4245E-01	1.4680E-02
9	4.6607E-02	1.4680E-02
10	1.5061E-01	1.4680E-02
11	5.1001E-03	2.1419E-02
12	6.8001E-01	2.1419E-02
13	2.4262E-02	1.2867E-03
14	1.5115E-01	1.2867E-03
15	1.3235E-02	1.2867E-03
16	4.0850E-04	1.2867E-03
17	5.6086E-02	1.2867E-03
18	5.6619E-02	1.2853E-03
19	6.8001E-02	2.1419E-02
20	4.0807E-04	1.2853E-03
21	1.2988E-02	1.2853E-03
22	1.5098E-01	1.2853E-03
23	2.4139E-02	1.2853E-03

AUX. FEED WATER RELIABILITY ANALYSIS.

TABLE - 6 :

 ***** TIME POINT DATA *****

THE NUMBER OF POINTS GENERATED = 23
 TOTAL TIME PERIOD IN HOURS (DEFAULT=1 YR) = 1440.000

THE TIME POINTS (IN HOURS) ARE PRINTED BELOW:

1 = 0.0	2 = 1.2000E+02	3 = 2.4000E+02	4 = 2.4150E+02	5 = 2.4570E+02
6 = 3.6000E+02	7 = 4.8000E+02	8 = 4.8150E+02	9 = 4.8570E+02	10 = 6.0000E+02
11 = 7.2000E+02	12 = 7.2150E+02	13 = 7.2570E+02	14 = 8.4000E+02	15 = 9.6000E+02
16 = 9.6150E+02	17 = 9.6570E+02	18 = 1.0800E+03	19 = 1.2000E+03	20 = 1.2015E+03
21 = 1.2057E+03	22 = 1.3200E+03	23 = 1.4400E+03		

Table C.2 Continued (sheet 10)

AUX. FEED WATER RELIABILITY ANALYSIS.

TABLE - 7 :

 ***** SYSTEM UNAVAILABILITY DATA *****

 (1) RUN OPTIONS:

SYSTEM DESCRIPTION	ACCURACY NUMBER	UNAVAIL. OPTION	PLOT OPTIONS X Y	CUTOFF OPTION
AUX. FEED WATER SYSTEM REL. ANALYSIS.	1	TOTL	NONE NONE	0

 (2) SYSTEM MEAN UNAVAILABILITIES BETWEEN ZERO AND 60.00 DAYS:

TOTAL MEAN UNAVAIL	UNAVAIL DUE TO FAILURES	% OF TOTAL	UNAVAIL DUE TO TESTING	% OF TOTAL	UNAVAIL DUE TO REPAIRS	% OF TOTAL
2.683E-04	1.712E-04	63.82	9.349E-05	34.85	3.589E-06	1.34

 (3) PEAK SYSTEM UNAVAILABILITIES:

POINT NUMBER	TIME (DAYS)	TIME (HOURS)	SYSTEM UNAVAIL
16	4.0063E+01	9.6150E+02	4.1869E-02
4	1.0063E+01	2.4150E+02	4.0636E-02
12	3.0063E+01	7.2150E+02	3.0064E-03
8	2.0063E+01	4.8150E+02	2.1356E-03
20	5.0063E+01	1.2015E+03	2.1338E-03
15	4.0000E+01	9.6000E+02	3.5911E-04
17	4.0237E+01	9.6570E+02	3.1884E-04
5	1.0237E+01	2.4570E+02	3.0925E-04
14	3.5000E+01	8.4000E+02	2.9956E-04
11	3.0000E+01	7.2000E+02	2.6689E-04
23	6.0000E+01	1.4400E+03	2.6684E-04
13	3.0237E+01	7.2570E+02	2.6589E-04

Table C.2 Continued (sheet 11)

TABLE - 8 ;

Table C.2 Continued (sheet 12)

SYSTEM UNAVAILABILITIES BETWEEN 0.0 AND 60.00 DAYS:									
POINT NUMBER	TIME (DAYS)	TIME (HOURS)	UNAVAIL AT T-EPSILON	UNAVAIL AT T-EPSILON	UNAVAIL AT T-EPSILON	UNAVAIL. INCREMENT	PERCENT OF TOTAL	AREA TYPE	
1	0.0	0.0	5.2883E-05						
2	5.0000E+00	1.2000E+02	1.0229E-04	1.0229E-04		6.4057E-06	2.4100		
3	1.0000E+01	2.4000E+02	1.5479E-04	1.5479E-04		1.0712E-05	3.9927		
4	1.0063E+01	2.4150E+02	4.0636E-02	4.0636E-02		4.2320E-05	15.7744		T
5	1.0237E+01	2.4570E+02	3.0925E-04	3.0925E-04		9.0149E-07	0.3360		R
6	1.5000E+01	3.6000E+02	1.0814E-04	1.0814E-04		6.5078E-06	2.4555		
7	2.0000E+01	4.8000E+02	1.6442E-04	1.6442E-04		1.1357E-05	4.2331		
8	2.0063E+01	4.8150E+02	2.1356E-03	2.1356E-03		2.2192E-06	0.8272		T
9	2.0237E+01	4.8570E+02	1.7000E-04	1.7000E-04		4.9294E-07	0.1837		R
10	2.5000E+01	6.0000E+02	2.0810E-04	2.0810E-04		1.4444E-05	5.3837		
11	3.0000E+01	7.2000E+02	2.6689E-04	2.6689E-04		1.9791E-05	7.3768		
12	3.0063E+01	7.2150E+02	3.0064E-03	3.0064E-03		3.1263E-06	1.1653		T
13	3.0237E+01	7.2570E+02	2.6585E-04	2.6585E-04		7.7504E-07	0.2880		R
14	3.5000E+01	8.4000E+02	2.9956E-04	2.9956E-04		2.1606E-05	8.0736		
15	4.0000E+01	9.6000E+02	3.5911E-04	3.5911E-04		2.7444E-05	10.2296		
16	4.0063E+01	9.6150E+02	4.1869E-02	4.1869E-02		4.3059E-05	16.2531		T
17	4.0237E+01	9.6570E+02	3.1864E-04	3.1864E-04		9.2802E-07	0.3462		R
18	4.5000E+01	1.0800E+03	1.1157E-04	1.1157E-04		6.7931E-06	2.5320		
19	5.0000E+01	1.2000E+03	1.6986E-04	1.6986E-04		1.1726E-05	4.3707		
20	5.0063E+01	1.2015E+03	2.1338E-03	2.1338E-03		2.2173E-06	0.8265		T
21	5.0237E+01	1.2057E+03	1.6996E-04	1.6996E-04		4.9283E-07	0.1837		R
22	5.5000E+01	1.3200E+03	2.0806E-04	2.0806E-04		1.4441E-05	5.3828		
23	6.0000E+01	1.4400E+03	2.5684E-04	2.5684E-04		1.9787E-05	7.3755		

SYSTEM MEAN UNAVAILABILITIES BETWEEN 0.0 AND 60.00 DAYS:					
TOTAL MEAN UNAVAIL	UNAVAIL DUE TO FAILURES	% OF TOTAL	UNAVAIL DUE TO TESTING	UNAVAIL DUE TO REPAIRS	% OF TOTAL
2.683E-04	1.712E-04	63.82	9.349E-05	3.589E-06	1.34

Table C.2 Continued (sheet 13)

*** AUX. FEED WATER RELIABILITY ANALYSIS.

NUMBER OF COMPONENTS = 23
 NUMBER OF TRIALS = 1200

*** COMPONENT VALUE

COMPONENT NO.	LAMDA	DIS	PARAMETER SPREAD & DISTRIBUTION				QRESID	DIS
			TC	DIS	TR	DIS		
1	0.0		0.0		0.0		30.000	L
2	0.0		0.0		0.0		10.000	L
3	0.0		0.0		0.0		10.000	L
4	0.0		0.0		0.0		10.000	L
5	0.0		0.0		0.0		10.000	L
6	0.0		0.0		0.0		10.000	L
7	10.000	L	3.000	L	3.000	L	10.000	L
8	10.000	L	3.000	L	3.000	L	10.000	L
9	0.0		0.0		0.0		10.000	L
10	10.000	L	3.000	L	3.000	L	10.000	L
11	0.0		0.0		0.0		30.000	L
12	0.0		0.0		0.0		3.000	L
13	10.000	L	3.000	L	3.000	L	10.000	L
14	0.0		0.0		0.0		10.000	L
15	10.000	L	3.000	L	3.000	L	10.000	L
16	0.0		0.0		0.0		10.000	L
17	10.000	L	3.000	L	3.000	L	10.000	L
18	10.000	L	3.000	L	3.000	L	10.000	L
19	0.0		0.0		0.0		10.000	L
20	0.0		0.0		0.0		10.000	L
21	10.000	L	3.000	L	3.000	L	10.000	L
22	0.0		0.0		0.0		10.000	L
23	10.000	L	3.000	L	3.000	L	10.000	L

Table C.2 Continued (sheet 14)

*** S AUX. FEED WATER RELIABILITY ANALYSIS.

DISTRIBUTED VALUES:

MEDIAN POINT VALUE	=	3.1497E-04
TRUE MEDIAN	=	5.9415E-04
ERROR FACTOR	=	7.2066E+00
MEAN	=	1.2970D-03
STANDARD DEVIATION	=	3.2431D-03
5% BOUND	=	1.3432E-04
95% BOUND	=	4.2816E-03
BETA 1	=	1.6376D+02
BETA 2	=	2.3341D+02
THIRD CENT. MOMENT	=	4.3648D-07
4TH CENT. MOMENT	=	2.5819D-08

*** \$ AUX. FEED WATER RELIABILITY ANALYSIS.

CONFIDENCE BOUNDS

CONFIDENCE	TOP EVENT VAL	INTERVALS	PROBABILITY	MAX. ERROR	MIN. PROBABILITY
0.5%	7.5631E-05	0.0%- 1.0%	9.8594E-01	1.0857E-01	8.7737E-01
1.0%	8.6380E-05	0.5%- 1.5%	9.1820E-01	6.8580E-02	8.4970E-01
2.5%	1.0510E-04	2.0%- 3.0%	7.3274E-01	5.5122E-02	6.7762E-01
5.0%	1.3432E-04	4.5%- 5.5%	5.7322E-01	4.6001E-02	5.2722E-01
10.0%	1.8046E-04	9.5%- 10.5%	4.3630E-01	3.6199E-02	4.0010E-01
15.0%	2.2543E-04	14.5%- 15.5%	3.7237E-01	3.1124E-02	3.4125E-01
20.0%	2.6124E-04	19.5%- 20.5%	3.3499E-01	2.8038E-02	3.0696E-01
25.0%	3.0705E-04	24.5%- 25.5%	3.1084E-01	2.5996E-02	2.8485E-01
30.0%	3.6497E-04	29.5%- 30.5%	2.9454E-01	2.4589E-02	2.6995E-01
40.0%	4.8247E-04	39.5%- 40.5%	2.7033E-01	2.2952E-02	2.5337E-01
50.0%	5.9415E-04	49.5%- 50.5%	2.7097E-01	2.2358E-02	2.4861E-01
60.0%	7.6224E-04	59.5%- 60.5%	2.7633E-01	2.2952E-02	2.5337E-01
70.0%	1.0068E-03	69.5%- 70.5%	2.9454E-01	2.4589E-02	2.6995E-01
75.0%	1.2071E-03	74.5%- 75.5%	3.1084E-01	2.5996E-02	2.8485E-01
80.0%	1.4576E-03	79.5%- 80.5%	3.3499E-01	2.8038E-02	3.0696E-01
85.0%	1.8216E-03	84.5%- 85.5%	3.7237E-01	3.1124E-02	3.4125E-01
90.0%	2.5920E-03	89.5%- 90.5%	4.3630E-01	3.6199E-02	4.0010E-01
95.0%	4.3103E-03	94.5%- 95.5%	5.7322E-01	4.6001E-02	5.2722E-01
97.5%	5.8253E-03	97.0%- 98.0%	7.3274E-01	5.5122E-02	6.7762E-01
99.0%	1.1847E-02	98.5%- 99.5%	9.1820E-01	6.8580E-02	8.4970E-01
99.5%	1.9284E-02	99.0%-100.0%	9.8594E-01	1.0857E-01	8.7737E-01

Table C.2 Continued (sheet 15)

Table C.2 Continued (sheet 16)

END OF CALCULATION BY UNRAC



Room 14-0551
77 Massachusetts Avenue
Cambridge, MA 02139
Ph: 617.253.5668 Fax: 617.253.1690
Email: docs@mit.edu
<http://libraries.mit.edu/docs>

DISCLAIMER OF QUALITY

Due to the condition of the original material, there are unavoidable flaws in this reproduction. We have made every effort possible to provide you with the best copy available. If you are dissatisfied with this product and find it unusable, please contact Document Services as soon as possible.

Thank you.

Pages are missing from the original document.

PAGE - 241