42
1-10-75

Dr-1761

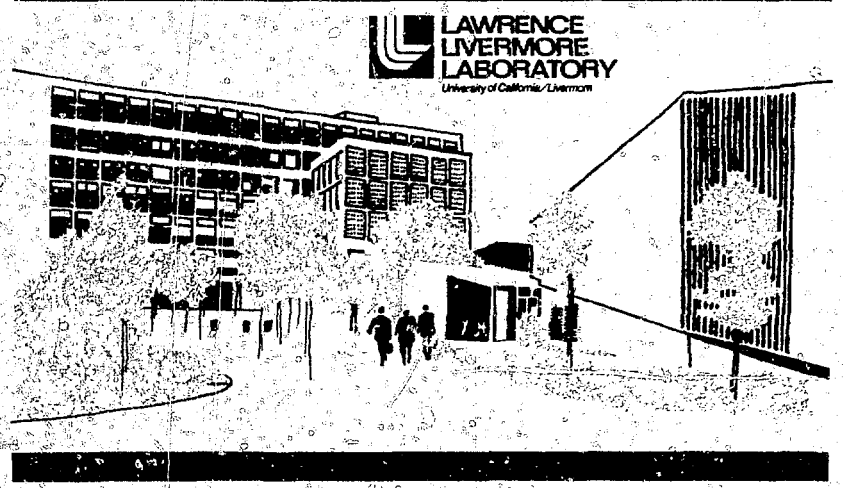
UCRL-51829

FAULT TREES FOR DECISION MAKING IN SYSTEMS ANALYSIS

Howard E. Lambert
(Ph. D. Thesis)

October 9, 1975

Prepared for U.S. Energy Research & Development Administration under contract No. W-7405-Eng-48



Printed in the United States of America
Available from
National Technical Information Service
U. S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy $___*; Microfiche $2.25

*Pages	NTIS Selling Price
1-50	$4.00
51-150	$5.45
151-325	$7.60
326-500	$10.60
501-1000	$13.60

Distribution Category
UC-32

LAWRENCE LIVERMORE LABORATORY
University of California/Livermore, California/94550

UCRL-51829

FAULT TREES FOR DECISION MAKING IN SYSTEMS ANALYSIS

Howard E. Lambert

(Ph. D. Thesis)

MS. Date: October 9, 1975

FAULT TREES FOR DECISION MAKING IN SYSTEMS ANALYSIS

Howard E. Lambert
University of California, Lawrence Livermore Laboratory
Livermore, California

ABSTRACT

New results in reliability theory pertinent to fault tree analysis are given. Concepts of probabilistic importance are described within the framework of fault tree analysis and applied to the areas of system design, diagnosis and simulation. The IMPORTANCE computer code ranks basic events and cut sets according to various measures of importance.

The application of fault tree analysis (FTA) to system safety and reliability is presented within the framework of system safety analysis. The concepts and techniques involved in manual and automated fault tree construction are described and their differences noted. The theory of mathematical reliability pertinent to FTA is presented with emphasis on engineering applications. An outline of the quantitative reliability techniques of the Reactor Safety Study is given.

New results in reliability theory pertinent to FTA include (1) an upper bound on the distribution of time to first failure, and a lower bound on the mean time to first failure for maintained systems and (2) an expression for the limiting unavailability of a component due to out-of-tolerance conditions.

Concepts of probabilistic importance are presented within the fault tree framework and applied to the areas of system design, diagnosis and simulation. The computer code IMPORTANCE which was developed by the author ranks basic events and cut sets according to a sensitivity

analysis. A useful feature of the IMPORTANCE code is that it can accept relative failure data as input. The output of the IMPORTANCE code can (1) assist an analyst in finding weaknesses in system design and operation, (2) suggest the most optimal course of system upgrade and (3) determine the optimal location of sensors within a system.

A general simulation model of system failure in terms of fault tree logic is described. The model is intended for efficient diagnosis of the causes of system failure in the event of a system breakdown. It can also be used to assist an operator in making decisions under a time constraint regarding the future course of operations. The model is well suited for computer implementation. New results incorporated in the simulation model include (1) an algorithm to generate repair checklists on the basis of fault tree logic and (2) a one-step-ahead optimization procedure that minimizes the expected time to diagnose system failure.

The methods developed are applied to aerospace, chemical and nuclear systems.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

IN DEDICATION

to my mother

and father

ACKNOWLEDGMENTS

There are numerous individuals and institutions who contributed greatly to my graduate education and made this thesis possible. I am truly indebted to the Lawrence Livermore Laboratory for sponsoring my doctoral research. During the course of my research at the Laboratory, I had the opportunity of working on the Reactor Safety Study as a fault tree analyst. In particular I wish to thank Ernie Hill and Walt Arnold of the Mechanical Engineering Department at the Laboratory. I owe to them the valuable experience that I gained in the field of reliability and fault tree analysis by conducting my research at Livermore. To these gentlemen, I am extremely grateful.

I am also grateful to the Departments of Nuclear Engineering and Industrial Engineering and Operations Research of the University of California at Berkeley. My sincere appreciation to my research advisers, Professor George Yadigaroglu and Professor Richard Barlow for guiding my research at Berkeley; I highly respect both of these individuals. I also wish to thank Professor Thomas Pigford for serving as the third member to my thesis committee. I considered working on an interdisciplinary thesis at Berkeley a unique experience. I had numerous contacts with people who were at the forefront of safety and reliability technology.

My sincere gratitude goes to the former Atomic Energy Commission and the present Energy Research and Development Administration, which supported me throughout my graduate education by granting me a fellowship for the first three years of my graduate education and, of course, in directly contributing to my support at Livermore.

I also wish to thank Jack Mansfield, Director of the School of Engineering at George Washington University (GWU). By being a participant lecturer in the short course, Fault Tree Analysis, at GWU, I gained practical insight into reliability engineering problems associated with various fields of engineering.

At this point, I wish to thank the individuals who in some way assisted me in my research. I had numerous conversations with Professor Jerry Fussell now at the University of Tennessee, concerning fault tree construction methodology. Jerry introduced me to some interesting research areas in the field of reliability pertinent to fault tree analysis. I also had numerous conversations with Dr. William Vesely at the Nuclear Regulatory Commission. Dr. Vesely made me aware of many practical engineering problems associated with reliability and fault tree analysis. I also wish to thank Dave Haasl, Institute of Systems Sciences, Inc., whose course introduced me to fault tree analysis.

At Lawrence Livermore Laboratory, I wish especially to thank Garth Cummings and Jack Karush. Garth read the first draft of the thesis and had several valuable comments regarding its content. Jack helped me formulate some of the concepts of probabilistic importance presented in the thesis.

Other individuals I wish to thank, listed by organization, include: Nozer Singpurwalla, George Washington University; Sheldon Ross, Davinder Sethi and Pradip Pande, Department of Operations Research, Berkeley; Guy Corynen, Fred Fritsch, Jim Wells, Bill Miller, Jack Savage, Don Thompson, Pat Gray, Al Cassell, Gail Dennis, Louise Green, the Technical Information Department and the Educational Policy Committee, Lawrence Livermore Laboratory; Lew Bass, Hans Wynholds and Bill Porterfield,

Lockheed Missiles and Space Company; Gus Wanner and Steve Wilson, General Electric Company; Arnie Rosenthal, University of Michigan; Ensup Yoon, Department of Chemical Engineering, Massachusetts Institute of Technology; Gary Powers, Department of Chemical Engineering, Carnegie Mellon University; Ernie Henley, University of Houston; Tom Smith, Battelle Northwest Laboratories; Jack Mansfield, Science Applications, Inc.; and Purnendu Chatterjee, Stanford Research Institute. A sincere note of apology to anyone omitted in the above list.

A special note of gratitude goes to Shirley Busey who typed the first draft and the final draft of the thesis. Her dedicated effort and skillful typing in the preparation of the thesis are sincerely appreciated.

Scope, Objective and Presentation of the Thesis

The author had the opportunity of attending a fault tree conference given in Berkeley, California, in September of 1974. The conference was attended by a diverse audience of engineers, statisticians and mathematicians. One evident fact surfaced during the panel discussions at this conference. A rather large gap exists between the elegant and elaborate mathematical methods of reliability theory and their application to the reliability engineering problems. The engineers claimed that the gap existed because the mathematicians did not concentrate on applying their elaborate theory to real world problems and because they did not bother to formulate a methodology for general applications. The mathematicians claimed that the engineers are not willing to take the time to study the mathematical theory of reliability and that engineers discard mathematical results for lack of understanding.

One goal of the author in this thesis is to bridge this gap. The thesis attempts to present the theory of mathematical reliability pertinent to fault tree analysis with an emphasis on engineering interpretations and applications. It shows what bounding procedures are necessary for making the solutions to real world problems tractable. It points out (1) the distinguishing features of systems currently being analyzed by fault tree analysis and (2) how the application of reliability calculations differs from system to system.

The main objective of the thesis is, however, to make fault trees a tool for decision making in systems analysis. We discuss below (by chapter) how this objective is accomplished.

In Chapter One, we put fault tree analysis in a system-safety perspective. We present system safety modeling techniques, including fault tree analysis, and show how they can be applied in a global safety anaysis in analyzing a system throughout its life cycle. We also describe the event tree methodology of the Reactor Safety Study [17]. We discuss the theory of manual and automated fault tree construction in detail, including the methodology of fault tree development at the top level. We attempt to make the reader aware of the engineering considerations and assumptions involved in the construction of the fault tree. We show one method for structuring fault trees that allows the inclusion of mutually exclusive fault events.

Chapter Two discusses the methods of probabilistic evaluation of fault trees in terms of coherent structure theory. It attempts to explain the concept of structural and statistical independence and how fault trees can be evaluated to allow for statistical dependency. New methods are proposed in (1) finding an upper bound on the distribution of time to first failure for a maintained systems and (2) finding the limiting unavailability of a component due to out-of-tolerance conditions. This chapter concludes by discussing the reliability quantification techniques of the Reactor Safety Study.

Chapter Three presents the theory of probabilistic importance and the mathematical expressions that are required to compute importance. The purpose of computing probabilistic importance is to generate a numerical ranking to assess weaknesses in a system. Such a ranking is analogous to a sensitivity analysis. The concept and application of probabilistic importance is the major contribution of this thesis. A key concept used in Chapter Three is the concept of proportional hazards.

This concept permits us to upgrade system designs on the basis of failure data that is relative rather than absolute in nature. The IMPORTANCE computer code presented in Appendix A computes various measures of probabilistic importance. The availability of such a code contributes to making fault tree analysis a design tool. For systems where repair is not allowed, the code accepts proportional hazards as input data. In another option, where repair is permitted, failure rate data can also be expressed in relative terms by representing the failure rate and repair rate data for the basic events in terms of a reference time unit. New computer algorithms are given in Appendix A that increase the computational efficiency of probabilistically evaluating fault trees.

In Chapter Four, we apply the concept of probabilistic importance to system design. A new expression called the upgrading function is given there that the author claims is the appropriate measure of importance in upgrading system designs. With the aid of new expressions developed in this chapter, we show how probabilistic importance can be calculated to determine the optimal location of sensors in a system.

In Chapter Five we show how the concept of probabilistic importance can be applied to the areas of system diagnosis and simulation and how repair checklists can be generated on the basis of fault tree logic. A one-step-ahead optimization procedure suitable for diagnosing a system under a time constraint is derived. We suggest options available to an operator when system fault conditions occur and how the future course of system operation can be determined on the basis of a risk assessment. In particular we consider the decision regarding shutdown at a nuclear power plant when a standby engineered safeguard system is found inoperable during plant operation.

CHAPTER ONE

SYSTEM SAFETY ANALYSIS AND FAULT TREE ANALYSIS

1.1 Introduction

Fault tree analysis, FTA, is an integral part of system safety analysis. System safety analysis is an analytical process that identifies and analyzes potential safety and reliability problems existing within a system. Reliability is a measure of the system's capability to function during the system's mission under prescribed specifications. Safety is concerned with the risk or danger posed to personnel or to the public when the system performs its task.

Chapter One describes FTA within a system safety context. It reviews and describes those methods including FTA that can be used in analyzing a system for reliability and safety. The theory of automated and manual fault tree construction is presented. A description of the fault tree methodology at the top level is given. The qualitative decisions that can be made once the fault tree is constructed are emphasized. Thus this chapter serves as an introduction to the theory of decision making on the basis of quantitative analysis of fault trees which is developed in later chapters.

1.2 Historical Aspects of Systems Analysis

A systems approach to reliability and safety evolved from the aerospace industry in the late 50's and the early 60's. At this time complex nuclear warhead missiles were being built that required analytical techniques capable of predicting accidents before their occurrence. In 1962 the Air Force adopted safety standards for ballistic missiles. In

1966 the Department of Defense (DOD) adopted the Air Force standards and required system safety in all phases of system development for all defense contracts. These standards were revised and in July of 1969 the DOD adopted MIL-STD-882 as the standard requirement for all defense contractors. [50], [60]

In 1965 the Boeing Company and the University of Washington sponsored a system safety symposium in Seattle, Washington [68]. It was recognized there that aerospace technology could be successfully extended to nuclear reactor safety technology and to various other commercial operations.

In 1967 Garrick et al [33] suggested a data collection program for nuclear power plant subsystems and components. They recommended implementing aerospace techniques in quantifying system reliability and safety and establishing the relative importance of various components to system operation.

In the mid 60's the United Kingdom Atomic Energy Agency, UKAEA, actually adopted a data collection program [36]. Farmer [20] from the UKAEA analyzed a spectrum of reactor accidents in order to determine the overall risk from nuclear power plant operation. He described reactor accident sequences in terms of event trees. The initiating event in the sequence considered by Farmer was a breach in the containment of a gas cooled reactor. By plotting the frequency of the accident versus the release of radioactivity from the accident, he could identify accidents with a high level of risk. Risk in this case was defined to be the product of two factors, (1) the probability of occurrence of the accident and (2) its consequence.

The most extensive risk assessment of nuclear power plant operation was completed in 1974 by the United State Atomic Energy Commission (US-AEC) [77]. WASH 1400, the Reactor Safety Study, also known as the Rasmussen Study, analyzed a vast spectrum of nuclear accidents, numerically ranked them according to their probability of occurrence and then assessed their potential consequences to the public. The fault tree technique, which had found widespread use in the aerospace industry, was selected as the basic analytical tool for this investigation. Event trees similar to those described by Farmer were used to organize and present accident scenarios.

Another industry plagued by the handling of hazardous substances is the chemical industry. Potential accidents were identified, particularly at refineries, that posed a risk to the public. Problems of reliability were also identified. The current trend in the chemical industry is to build large, continous, single-line plants. Failure of equipment anywhere in these plants could shutdown the entire plant, causing considerable financial losses. In the early 1970's system safety and reliability techniques were also applied in the chemical industry. [2], [11], [53], [57].

1.3 Basic Concepts of Systems Analysis

The systems approach is a methodical concept in analyzing a system for reliability and safety. The approach emphasizes one important premise -- identification of hazardous conditions and problem areas during the conceptual and design stages of a system can prevent costly retrofits, unscheduled shutdowns and accidents during system operation.

Systems analysis is a directed process for the orderly acquisition of specific information pertinent to a given system. In particular we are interested in events that might cause injury or harm to people, damage to or loss of equipment or property, or interruption of work. Time and budget constraints require that we limit the scope of our investigation to some defined boundary. If the elements within the boundary have some significant relationship to one another, then in essence we have bounded the system. At this point it is instructive to define what is meant by a system. Levens [48] defines a system as an orderly arrangement of interrelated components that act and interact to perform some task or function in a particular environment and within a particular time period. Haasl [39] defines a system as an entity comprised of an interacting set of discrete elements. Grose [37] defines a system as any complete entity consisting of hardware, software, personnel, data, services and facilities which transforms known inputs into desired outputs.

For purposes of analysis, the system should be specified in terms of (1) its functional purpose, which specifies its task(s), the time period involved, and the environmental conditions; (2) its component constituency, which identifies subsystems, components, and people involved; and (3) the functional order of the system, which includes the interrelationships between components and subsystems and the information flow within the system (such as inputs, outputs, and logic).

To gain a detailed understanding of how a system may fail, we first must understand how it functions. Preparing a narrative functional description of a system and components for each operational mode of the

system is a good approach. Such a description should include enough detail to show the uniqueness and relevance of the functions performed.

A diagram of the system, showing all components, is also helpful. One method is to break the system down into major blocks, showing diagrammatically how the components interact to perform the function of each block, and depicting any interfaces that exist between blocks. This helps us to visualize all important interrelationships and simplifies tracing any malfunctions that propagate through the system. Other system diagrams include installation drawings, logic diagrams. piping and instrument diagrams and process flow sheets.

However, diagrams tend to limit the analyst's view to two dimensions. It is important for the analyst to visualize the system in three dimensions and to assess potential hazards associated with equipment proximity.

There are basically three sources of system information: (1) experience, both direct and related; (2) tests, simulation and confirmation; and (3) analysis. The information from direct experience is the most accurate but the most costly. Destructive or nondestructive tests on system elements are less expensive to perform than testing the entire system. However, as more and more basic system components are tested, the results lose validity, since the tests must be conducted out of final context. Analysis is the least accurate. However, analysis can direct testing and make it more effective. As system costs increase we place more dependence on analysis.

Direct experience in the nuclear power industry can be obtained from actual plant operating data, related experience from news releases and bulletins from the United States Nuclear Regulatory Commission, NRC.

WASH 1400 [77] and the United States Atomic Energy Commission Office of Operations Evaluation [76] recently compiled the NRC operating data and noted those incidents that had a major effect on nuclear power plant safety and availability. Holmes and Narver [34] will soon release a report that will cite factors having a major effect on fast reactor availability. Maintenance logs at a nuclear power plant can serve as a source of information at the component level. Component failure data can also be obtained from WASH 1400. The study compiled data from 1972-73 operating experiences. They assessed the data to a 90% confidence range, compared it to other industrial data sources and in general found no substantial disagreement. Human performance data is also included in WASH 1400.

In general there is a vast amount of experience in the chemical industry. Information at the systems level for a chemical process can be obtained from plant experience if the process if well known. For new processes that are still in the design stages, the first source of systems level information is process and maintenance data from the pilot plant or semiworks. Information at this level can also be acquired from similar chemical processes. Data at the component level in the chemical industry can be obtained from the SYREL data bank [67] which is an integral part of the UKAEA Systems Reliability Service.

1.4 Methods of Analysis

There are two formalized methods in system safety and reliability, inductive and deductive analysis. Inductive analysis involves postulating a possible state of components and/or subsystems and determining its overall effect on the system. Two basic inductive analysis

techniques are the preliminary hazards analysis (PHA) and the failure modes and effects analysis (FMEA). Other types of inductive analyses include decision trees or event trees and Markovian analysis.

Deductive analysis, on the other hand, takes an opposite approach. It involves postulating a possible state of the overall system and identifying those component states that may contribute to its occurrence. An example of deductive analysis is fault tree analysis (FTA).

1.5 Preliminary Hazards Analysis (PHA)

A PHA is a broad, all-encompassing study performed at the conceptual stages of the system design. Its objectives are to identify hazardous conditions inherent in a system and to determine the effect of any potential accidents. A major goal of PHA is to prevent accidents that have occurred in identical or similar systems.

The first step in PHA is to identify elements in hardware or functions that are inherently hazardous. As shown in Figure 1.1, these hazardous elements may be categorized by checklists as either hazardous energy sources of hazardous process or events. Hazardous energy sources are hazardous by themselves if released in the system environment. Hazardous processes or events are either physical or chemical processes that produce a hazardous condition when they interact with the system. Each company should compile a list of all basic hazards associated with its products. This list should be used as a checklist in performing a PHA to ensure all hazards have been identified.

Powers and Tomkins [57] identify two primary sources of hazards in the chemical industry. The first source includes the intrinsic properties of the materials in and around the process. These properties include the flammability, corrosiveness, reactivity, and toxicity of the

species in the process system. The second source includes hazards associated with equipment in the process, such as pressure vessels and chemical reactors.

Hazardous Energy Sources

1. Fuels
2. Propellants
3. Initiators
4. Explosive charges
5. Charged electrical capacitors
6. Storage batteries
7. Static electrical charges
8. Pressure containers
9. Spring-loaded devices
10. Suspension systems
11. Gas generators
12. Electrical generators
13. rf energy sources
14. Radioactive energy sources
15. Falling objects
16. Catapulted objects
17. Heating devices
18. Pumps, blowers, fans
19. Rotating machinery
20. Actuating devices
21. Nuclear devices
etc.

Hazardous Processes and Events

1. Acceleration
2. Contamination
3. Corrosion
4. Chemical dissociation
5. Electrical
   - shock
   - thermal
   - inadvertent activation
   - power source failure
   - electromagnetic radiation
6. Explosion
7. Fire
8. Heat and temperature
   - high temperature
   - low temperature
   - temperature variations
9. Leakage
10. Moisture
    - high humidity
    - low humidity
11. Oxidation
12. Pressure
    - high pressure
    - low pressure
    - rapid pressure changes
13. Radiation
    - thermal
    - electromagnetic
    - ionizing
    - ultraviolet
14. Chemical replacement
15. Mechanical shock
etc.

FIG. 1.1 *Checklists of Hazardous Sources* [16], [41], [49]

Experience in the aerospace industry and the aircraft industry indicates that accidents occur, often not as a result of a single random event, but as the result of a dynamic sequence of events which together

generate a specified outcome. The second step in a PHA is to identify the series of triggering events, i.e., causative factors, that can transform the hazardous element into a hazardous condition and in turn into a potential accident. The triggering events can be conditions, undesired events, or faults existing within the system.

It is common in the aerospace industry to rank hazards according to their effects. Class I hazards have negligible effects, Class II have marginal effects, Class III have critical effects and Class IV have catastrophic effects.

The next step in a PHA is to decide on the accident prevention measures that must be taken (particularly with Class III and IV hazards). Two courses of action are available: (1) corrective action in the form of equipment design changes, procedural changes, or redirection of mission goals; or (2) contingency action in the form of design of reactive protective system or training of personnel. Examples of protective systems in the chemical industry are sprinkler systems, fire walls, emergency cooling systems, explosion limiting devices, etc. Powers and Tomkins [57] define this as the protective-systems approach.

A common format for a PHA is a columnar form with specific entries. A sample PHA using this format appears in Figure 1.2.

A PHA should be a dynamic coordinated effort of many individuals. It should be updated, revised and expanded throughout the system life cycle. It should identify hardware failures requiring FMEA and events requiring FTA.

A PHA should also identify potential interface conditions, particularly where associated contractors design and build individual subsystems. The aerospace industry has been plagued with numerous accidents

Hazardous element	Triggering event 1	Hazardous condition	Triggering event 2	Potential accident	Effect	Corrective measures
1. Strong oxidizer	Alkali metal perchlorate is contaminated with lube oil	Potential to initiate strong redox reaction	Sufficient energy present to initiate reaction	Explosion	Personnel injury; damage to surrounding structures	Keep metal perchlorate at a suitable distance from all possible contaminates
2. Corrosion	Contents of steel tank contaminated with water vapor	Rust forms inside pressure tank	Operating pressure not reduced	Pressure tank rupture	Personnel injury; damage to surrounding structures	Use stainless steel pressure tank, locate tank at a suitable distance from equipment and personnel

FIG. 1.2 Format for Preliminary Hazards Analysis

1. Hazardous Situation - Alkali metal perchlorate is contaminated by a spill of lube oil.
2. Hazardous Situation - Moisture inside pressurized steel tank.

caused by unchecked system interface conditions. Rogers [60] cites the classic example that occurred in the early stages of the U.S. ballistic missile development. Four major accidents occurred as the result of numerous interface problems. In each accident, the loss of a multi-million dollar missile/silo launch complex resulted.

The failure of Apollo 13 was due to a subtle interface condition. [21], [35] During prelaunch, improper voltage was applied to the thermostatic switches to the heater of oxygen tank #2. This caused Teflon on the wires leading a fan inside the tank to crack. During flight, the switch to the fan was turned on, a short circuit resulted that caused Teflon to ignite and in turn caused the oxygen tank to explode.

WASH 1400 included in its risk assessment, human maintenance and testing interfaces on critical emergency systems and in many cases

identified a higher contribution to system failure from these sources than from hardware failures.

Thus, it may be emphasized that identification of potential interface conditions should be an integral part of a PHA.

Once the PHA is completed, the number of catastrophic and critical hazards indicates the magnitude and complexity of the safety problems associated with the proposed system. It is also a good indication of how much management attention is required to minimize or control these hazards.

1.6 Failures Modes and Effects Analysis (FMEA)

A failure modes and effects analysis is a detailed inductive analysis performed at the design stages of a system. It systematically analyzes all contributory component failure modes and identifies the resulting effect on the system. The purpose of FMEA is to identify areas in the design or hardware where improvements are required to ensure the system will be reliable and safe for its intended use.

The person most capable of performing a FMEA is the system design engineer most familiar with the subsystem or system. The system design engineer must first know all significant failure modes of each component comprising the subsystem or system. The four basic component failure modes are: (1) premature operation, (2) failure to operate at prescribed time, (3) failure to cease operation at a prescribed time and (4) failure during operation.

After all the significant failure modes of each of the system components are determined, the effect of each failure mode on the other system components and the effect on the overall performance of the

system with respect to the system's task are determined. A hazards classification is then assigned as in a PHA to reveal the severity of each component failure mode on the system. A description of the methods by which the occurrence of the failure modes of the different components can be detected could also be included in the FMEA. A suggested format for a FMEA is given in Figure 1.3b. The component analyzed is a low pressure injection pump designed to inject cooling water into the core of a pressurized water reactor, PWR, in the event of a loss of coolant accident (LOCA). The low pressure injection system, a standby safety cooling system, is shown in Figure 1.3a.

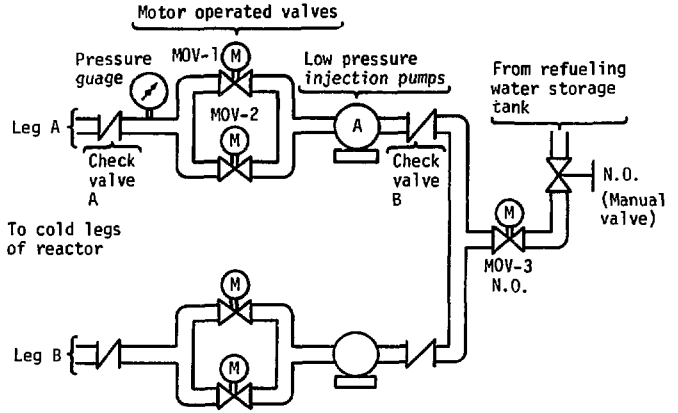


FIG. 1.3a Low Pressure Injection System [70]

The format below is suggested by Hammer [41]. Other formats are given in references [45] and [83].

Item	Failure mode	Cause of failure (internal)	Possible effects	Probability of occurrence	Criticality	Possible action to reduce failure rate of effects
Low pressure injection pump, LEG A	Fails to start	a. Bearing failure b. Insulation failure c. Brush failure	In event of LOCA, loss of redundant cooling capability	1×10^{-3}/demand (given in WASH 1400)	Marginal	Technical specifications by NRC require testing of emergency cooling system once a month

FIG. 1.3b Sample Format For FMEA

A critical items list results from the FEMA to reveal what components are critical to the system. If the failure rates of these components are known, then a criticality analysis (CA) is performed to show quantitatively the effect of each component failure on the system. The CA computes for each component a criticality number C_r, (see Section 4.2) that is a quantitative indication of the importance of the component to system operation.

If a component of high criticality or importance has to be retained in the system, then design changes that will reduce or eliminate component criticality are incorporated whenever feasible. These design changes produce corresponding changes in the critical items list. If at this point some components are still critical, a component-design engineer incorporates design changes in critical components through such means as part redundancy, part derating, and redesign to fail safe. If the final critical-items list still contains critical components, then quality control puts special controls, e.g., checking and maintenance, on these critical components.

The relative monetary value of design changes either at the system or component level can be determined by a cost-effective analysis. In

cost effective analysis, the cost of system changes made to increase safety are compared either with the decreased cost from fewer failures or with the increased effectiveness of the system to perform the task.

1.7 Markov Analysis

Failure Modes and Effects Analysis is a single-thread inductive analysis, i.e., the effect of each component state of the system is considered independently. Markovian analysis, on the other hand, considers multiple effects and is a multi-thread inductive analysis. This process can be used for operational simulations; however, the complexity of the analysis makes hand calculations impractical, and the performance of accurate simulations requires expensive equipment. Consult references [3], [59] and [65] for a discussion of Markov analysis and its application to engineering problems.

In a Markov process, all the mutually exclusive system states must be identified. The set of possible states in which the system is working is called the "good" set as opposed to the set of possible states in which the system is out of order, which is called the "bad" set. Of particular interest in the application of the Markov process is the determination of the probability of a system making a transition from the "good" set to the "bad" set as a function of time. Two restrictions apply, however, in the use of the Markov process: the system as it enters each state is influenced by what has happened in the immediately preceding state only and does not depend on any other previous system states. Another restriction is that the rates of system transition among possible states must be constant with respect to time to make the problem tractable.

As an example of the Markov process, consider a system of two units. For the moment assume we are not interested in the logical connection of these units. Each component is assigned a separate repair team to restore a failed unit to a good-as-new state. We assume each unit to be in one of two mutually exclusive states (1) the unit is operated as intended or (2) it is failed and under repair. In this case there are four mutually exclusive system states:

0: both units operation

1: unit one down and under repair, unit two up

2: unit two down and under repair, unit one up

3: both units down and under repair.

We can define $a_{ij}\Delta t$ as the conditional probability of the system making a transition from state i to state j in the time interval (t, t+Δt). The probability that the system remains in state i for (t, t+Δt) can be defined to be $(1-a_{ii})\Delta t$ where $a_{ii} = -\sum_{j \neq i}(1-a_{ij})$. Further define $P_i(t)$ to be the probability that the system is in state i at time t. An expression for the time rate of change of $P_i(t)$ can now be written. For example, for State 0

$$P_0(t + \Delta t) = P_0(t)\,[1 - (a_{01} + a_{02})\Delta t] + P_1(t)a_{10}\Delta t$$

$$+ P_2(t)a_{20}(t)\Delta t + O(\Delta t)^2,$$

where the first term on the right hand side of the above equation can be recognized as the probability the system remains in the state 0, the second and third terms as the probability of one unit being repaired in Δt, and the fourth term, the second order effect of simultaneously repairing both units in Δt (such as a transition from state 3 to state 0).

Neglecting second order effects, dividing by Δt and letting $\Delta t \to 0$ yields

$$\frac{dP_0(t)}{dt} = -(a_{01} + a_{02})\, P_0(t) + a_{10}P_1(t) + a_{20}P_2(t).$$

Equations for states 1, 2, and 3 can similarly be written

$$\frac{dP_1(t)}{dt} = a_{01}P_0(t) - (a_{10} + a_{13})\, P_1(t) + a_{31}P_3(t)$$

$$\frac{dP_2(t)}{dt} = a_{02}P_0(t) - (a_{20} + a_{23})\, P_2(t) + a_{32}P_3(t)$$

$$\frac{dP_3(t)}{dt} = a_{13}P_1(t) + a_{23}P_2(t) - (a_{31} + a_{32})P_3(t).$$

We can write the above equations in matrix form as

$$\frac{d\,\underline{P}\,(t)}{dt} = A\,\underline{P}\,(t)$$

$$= \begin{bmatrix} -(a_{01}+a_{02}) & a_{10} & a_{20} & 0 \\ a_{01} & -(a_{10}+a_{13}) & 0 & a_{31} \\ a_{02} & 0 & -(a_{20}+a_{23}) & a_{32} \\ 0 & a_{13} & a_{23} & -(a_{31}+a_{32}) \end{bmatrix} \begin{bmatrix} P_0(t) \\ P_1(t) \\ P_2(t) \\ P_3(t) \end{bmatrix}$$

and identify A to be the transition matrix. The above process can be represented diagramatically as in Figure 1.4.

To solve the above coupled first order differential equations, the system transition rates must be known. The rate of breakdown for a component in the literature is commonly referred to as λ, the failure rate.

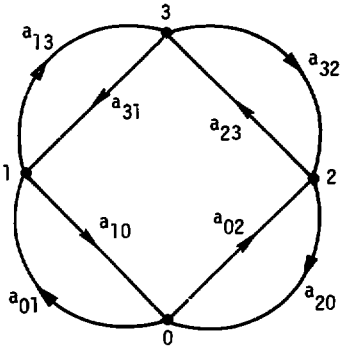


FIG. 1.4 States and Transition Rates For A System with Two Dissimilar Units and Two Repairmen

The rate of repair can be designated as ν. We can identify the above transition rates, a_{ij}, to be

$$a_{01} = a_{23} = \lambda_1$$

$$a_{02} = a_{13} = \lambda_2$$

$$a_{32} = a_{10} = \nu_1$$

$$a_{31} = a_{20} = \nu_2$$

where the subscripts on the repair and failure rates refer to the indicated unit.

The general solution to $\underline{P}(t)$ is a weighted sum of exponentials as shown below [3]

$$\underline{P}(t) = \begin{bmatrix} P_1(t) \\ P_2(t) \\ \vdots \\ P_n(t) \end{bmatrix} = \sum_{i=1}^{n} c_i \begin{bmatrix} A_{i,1} \\ A_{i,2} \\ \vdots \\ A_{i,n} \end{bmatrix} e^{\omega_i t}.$$

The powers to these exponentials, $\underline{\omega}$, are the eigenvalues corresponding to the transition matrix and n is the number of components. The number of absorbing states in the transition matrix is equal to the number of eigenvalues that are zero; the remaining are negative real constants. The vector

$$\begin{bmatrix} A_{i,1} \\ A_{i,2} \\ \vdots \\ A_{i,n} \end{bmatrix}$$

is one of the n eigenvectors corresponding to the transition matrix. Knowledge of $\underline{P}(t)$ at one point in time, e.g., P(0), determines $\underline{c}$ and results in a unique solution for $\underline{P}(t)$.

If the two units in the example are connected in parallel, the system is up if it is in states 0, 1 or 2. The probability that the system is up at time t, called the system availability, $A_s(t)$, is given by

$$A_s(t) = P_0(t) + P_1(t) + P_2(t);$$

if the system is connected in series the system availability is given by

$$A_s(t) = P_0(t).$$

In case of two units in parallel, if we disallow transitions from state 3, i.e., we make state 3 an "absorbing" state by setting $a_{31} = a_{32} = 0$, we can find the probability that the system has not failed by time

t, $F_s(t)$, also called the system reliability. In the series case repair has no effect on system reliability. In general the system reliability is less or equal to the system availability.

Where the effects of system failure are catastrophic, it is of interest to know the system reliability as a function of time, also called the distribution of time to first failure. In general for complex systems where repair is allowed, the time-dependent system reliability is a very difficult quantity to compute. Due to the large number of system states, the transition matrix is in turn large, making the Markov solution intractable. Also the Markov process cannot be used when the failure and repair rates are not constant in time. In Chapter Two, upper bounds to the distribution of time to first failure for general repair and failure distributions are given. For simple systems, these bounds can be compared with the Markov solution.

1.8 Event Trees

An event tree is an inductive logic diagram. The diagram starts with a given initiating event and depicts various sequences of events leading to multiple-outcome states. To each state is associated a particular consequence. The event-tree approach is similar to decision tree methodology in business applications. [77].

WASH 1400 used the event-tree methodology as the principal means of identifying significant sequences associated with nuclear power plant accidents. It also provided the necessary framework for the overall risk assessment by (1) providing a basis in defining accident scenarios for each initiating event, (2) by depicting the relationships of success and failure of safety related systems associated with various accident

consequences, and (3) by providing a means for defining top events to system fault trees.

The accidents considered in the Reactor Safety Study provide an excellent basis on which to describe the event-tree methodology in the context of a risk assessment.

A major goal of the Reactor Safety Study was to determine the risks to the public from commercial nuclear power plant operation. A potentially significant risk from these plants to the public is the release of substantial amounts of radioactivity. The vast amount of radioactivity at a nuclear power plant is stored as fission products contained in the ceramic UO_2 fuel located in the core of the reactor. To release this radioactivity in significant amounts, the UO_2 fuel must be heated to its melting point. This can occur as the result of the interruption of heat flow from the UO_2 fuel to the heat sink. One way this can occur is the loss of heat removal capability caused by a breach of the pressure boundary of the primary cooling system. If the emergency cooling systems do not operate during the loss of coolant accident (LOCA), and the containment enclosing the reactor vessel does not effectively contain the fission products, a major release of radioactivity results. A simplified schematic of the layout of the emergency cooling systems utilized for the injection mode following the LOCA is given, for a pressurized water reactor (PWR) in Fig. 1.5. The injection mode for these systems takes place for a period of approximately thirty minutes following the LOCA, when water from the refueling water storage tank is discharged through the injection pumps P1, P2, P3 and P4. There is a spectrum of accidents that can result in smaller releases. The simplified event tree in Figure 1.6 depicts this idea.

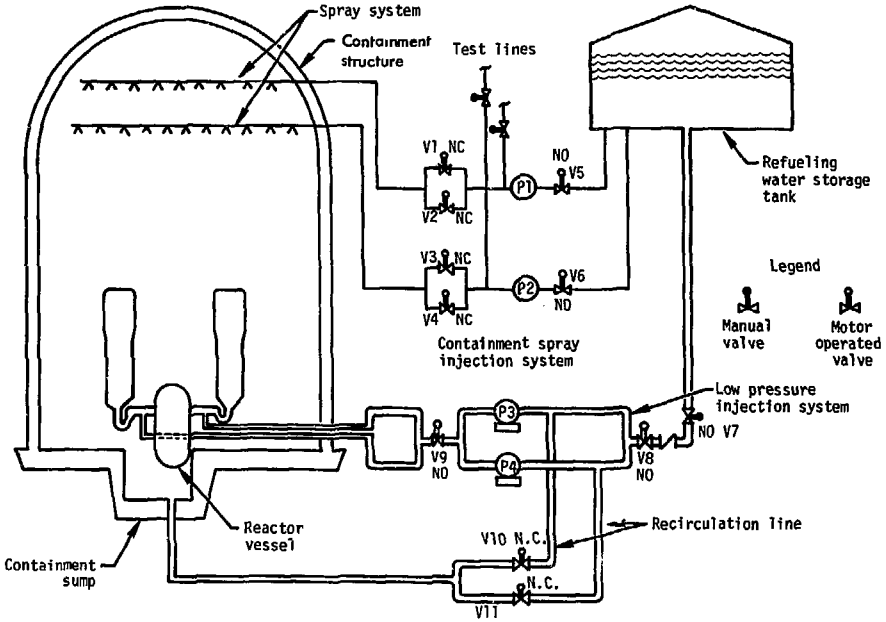


FIG. 1.5 Containment Spray Injection and Low Pressure Injection and Recirculation Systems

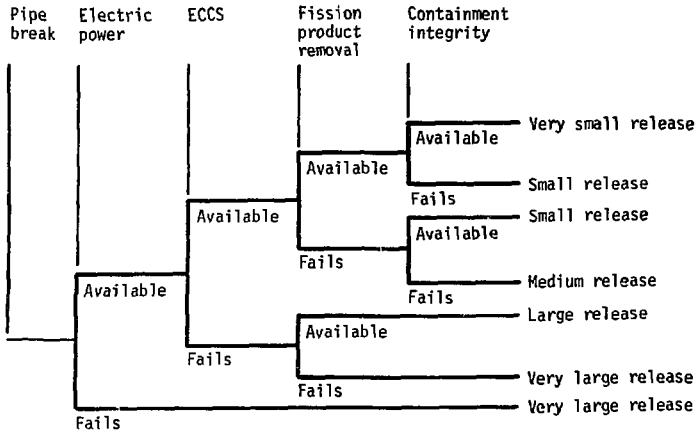


FIG. 1.6 Simplified Event Tree for a LOCA in a Typical Nuclear Power Plant

The initiating event considered is a pipe break in a coolant loop. At the first branch, the status of electric power is considered. If it is unavailable, as indicated by the downward step, a major release results since all the systems are inoperable without electric power. If electric power is available, the next event to be considered is the status of the emergency core cooling system (ECCS). Failure of the ECCS results in a fuel meltdown and eventual breach of the containment. If the containment spray system is working, it will remove some fission products. The release of radioactivity in this case is not as great as if the spray system is not working. As shown by the top branch of the

event tree, if all engineered safety systems, ESS, operate as intended, only a very small release results.

The event tree as described demonstrates the thought process involved in its development. In general, the event trees considered in the study were much more complex. The event trees had to consider functional interdependencies, cooling requirements and questions of partial failure. The mode of containment failure depended upon the availability of the ESS with respect to time, and upon the physical processes involved such as the rate of fuel melting, missiles from steam explosions, hydrogen combustion from Zr-H_2O reactions and CO_2 generation from decomposition of concrete.

The complex event trees in the study produced an enormous number of sequences to be considered. However, functional interdependencies eliminated many system failures from further consideration. For example, with electric power unavailable, the status of the entire engineered safeguard system is irrelevant. Also, timing considerations eliminated many sequences. The status of the ECCS during the recirculation mode* is immaterial if the ECCS failed during the initial injection mode.

As shown above, three factors dominated in the generation of accident sequences from the event trees; schematically

$$\text{Accident Sequence} = \underset{\text{Event}}{\text{Initiating Event}} \times \underset{\text{Failure}}{\text{System}} \times \underset{\text{Failure Mode.}}{\text{Containment}}$$

Initiating events considered other than pipe breaks were transient events and the catastrophic rupture of the pressure vessel. Each

*After an initial injection period, the ECCS recirculates the injected water that is collected at the sump of the containment building.

defined system failure from the accident sequence served as a top event of a fault tree which was then constructed for the particular system. The containment failure mode in each sequence was the major factor in determining the amount, composition and timing of the release. The Battelle CORRAL computer code [75] determined the isotopic composition and amount of radionuclides released from various accident chains following the accident. Accident sequences were then grouped into representative release categories suitable for consequence modeling.

The collection of probabilities and consequences for the various accident chains gave the required points from which the probability-versus-release histograms can be plotted. The consequence modeling considered fatalities, injuries, long-term health effects, and property damage.

Section 2.8 discusses how probabilities for accident chains can be calculated to allow for dependencies and in particular how the system fault trees can be quantified to allow for various "common mode" contributions.

1.9 Fault Tree Analysis (FTA)

1.9.1 Introduction - Fault Tree Analysis is a formalized deductive analysis technique that provides a systematic approach to investigating the possible modes of occurrence of a defined system state or undesired event. Fault tree analysis, FTA, was first conceived by H. A. Watson of Bell Telephone Laboratories in connection with an Air Force contract to study the Minuteman missile launch-control system. Boeing Company analysts extended the technique and developed computer programs for both qualitative and quantitative analysis. It was recognized that fault

that fault tree analysis could be successfully extended from the aerospace technology to nuclear reactor reliability, safety, and availability technology, and to various other commercial operations such as the chemical processing industry.

Undesired events requiring FTA are identified either by inductive analysis, such as a preliminary hazard analysis, or by intuition. These events are usually undesired system states that can occur as a result of subsystem functional faults. These events can be broad, all-encompassing events, such as "Release of Radioactivity from a Nuclear Power Plant" or "Inadvertent Launch of an ICBM Missile," or they can be specific events, such as "Failure to Insert Control Rods" or Energizing Power Available on Ordinance Ignition Line".

FTA consists of two major steps, (1) the construction of the fault tree and (2) its evaluation. The evaluation of the fault tree can be qualitative, quantitative, or both depending upon the scope and extensiveness of the analysis.

The objectives of fault tree analysis are: (1) to identify systematically all possible occurrence of a given undesired event, (2) to provide a clear and graphical record of the analytical process, and (3) to provide a baseline for evaluation of design and procedural alternatives. An introduction to FTA is given in this section. The reader should consult references [13], [23], [24], [38], [39], [47], [58], and [87] for a general discussion of FTA.

1.9.2 Fault Tree Construction - Fault tree construction has been discussed in references [15], [28], [38], and [47]. Some important considerations are given below.

1.9.2.1 Preliminary Considerations - The goal of fault tree construction is to model the system conditions that can result in the undesired event. Before the construction of the fault tree can proceed, the analyst must acquire a thorough understanding of the system. In fact, a system description should be part of the analysis documentation. The analyst must carefully define the undesired event under consideration, called the "top event". To make his analysis understandable to others, the analyst should clearly show all the assumptions made in the construction of the fault tree and the system description used. Practical considerations require that he scope the analysis, setting spatial and temporal bounds on the system. He should determine the limit of resolution, identify potential system interfaces and realize the constraints of the analysis in terms of the available resources, time and money.

1.9.2.2 Event Description - A fault tree is a deductive logic model that graphically represents the various combinations of possible events, both fault and normal occurring in a system that lead to the top event. The term "event" denotes a dynamic change of state that occurs to a system element. If the change of state is such that the intended function occurs as designated, the event is then a normal system function or normal event. If the change of state is such that the intended function of the particular element is not achieved or an unintended function is achieved, the event is an abnormal system function or fault event. Stated in other terms, normal events are events that are expected to occur and fault events are those that are not expected to occur. Fault events may be classified according to two types, type I; a system element fails to perform an intended function and

type II; a system element performs an inadvertent function. Examples of normal events include

(1) Battery removed for routine maintenance during system operation.

(2) Control rods are inserted when an operator pushes a scram bar.

Examples of type I fault events include

(1) Diesel generator fails to start when emergency bus voltage is lost.

(2) Pumps fail to start when switch is closed.

(3) Motor seizes during operation.

Examples of type II fault events include

(1) Spurious scram of reactor during operation.

(2) Electromagnetic energy energizes ordinance ignition line.

(3) Motor starts after system shutdown.

A fault is some component state-of-existence (not necessarily a failure) that contributes to a possible mode of occurrence of the undesired event. A failure is an inherent state of a system element in which the element is unable to perform its intended function. System elements include hardware, software, human and environmental conditions.

In order to apply Boolean logic in FTA, the outcome of each event must exhibit two states only, the OFF state and the ON state. The OFF state corresponds to an unfailed state for a system element. The ON state for a type I fault event corresponds to a failed state; for a type II fault event, the ON state corresponds to a state in which system elements are operating inadvertently. The ON state for a normal event corresponds to a normal operating state for a system element. A system element may return from the ON state to the OFF state because of repair, another fault event, or other factors relating to system design

and operation, such as shutdown of the system. The time at which a system element is ON is referred to as the fault duration time (FTD) for fault events and event duration time (EDT) for normal events. In the context of maintenance, components that are repairable have a finite fault duration time. The FDT may be of extreme importance to the analyst or design engineer. For example, consider two redundant components sharing a common load. While failure of one of these components may not in itself cause the system to fail, the FDT may determine the amount of safety degradation incurred until the failure is detected and corrected.

1.9.2.3 Event Symbols - The symbols shown in Figure 1.7 represent specific types of fault and normal events in fault tree analysis. The rectangle defines an event that is the output of a logic gate. Logic gates are discussed in the following paragraph. The circle defines a basic inherent failure of a system element when operated within its design specifications. It is, therefore, a primary failure, and is also referred to as a generic failure. The diamond represents a failure, other than a primary failure that is purposely not developed further. The house represents an event that must occur or is expected to occur because of design and normal conditions, such as a phase change in a system. A house can be used as a switch that is turned on and off during the course of the analysis. A house can represent a state input. For example, the Reactor Safety Study used a house to represent the location of a pipe break in a boiling water reactor. The house is a switch that is turned on with probability one during its effective duration otherwise it is turned off.

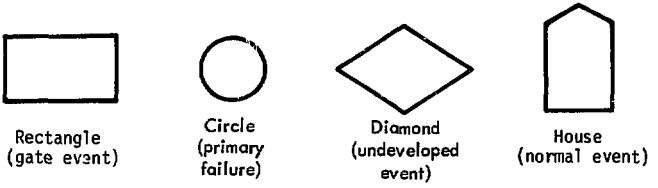


FIG. 1.7 Event Symbols

1.9.2.4 Logic Gates - The fundamental logic gates for fault tree construction are the OR and the AND gate. The OR gate describes a situation where the output event will exist if one or more of the input events exists. The AND gate describes the logical operation that requires the coexistence of all input events to produce the output event. The symbols for the logic gates are shown in Figure 1.8.

As an example of AND gate developments, consider the simple series circuit controlling a motor shown in Figure 1.9. The fault tree in Figure 1.10 identifies two basic hardware failures: switch 1 fails to open and switch 2 fails to open. We assume that in System A the wires or connectors do not contribute to the system failure.

Figure 1.11 illustrates an example of OR gate development. In this case, a fault tree is shown with top event "Motor does not start" for system A of Figure 1.9. The assumptions and initial conditions given in Figure 1.9 apply to Figure 1.11. We see in Figure 1.11 that the motor can fail to start if either event 1, "motor fails to start", occurs or event 2, "circuit fails to supply current to motor", occurs. Event 1 represents a failure of the motor due to internal causes when operated

within its design envelope and is a basic event. Event 2 is not a basic cause and must be developed further.

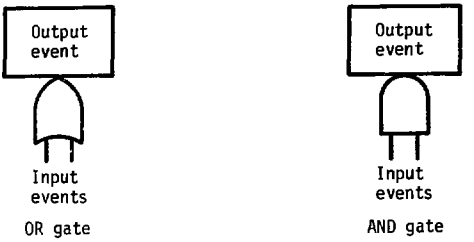


FIG. 1.8 Symbols for Logic Gates

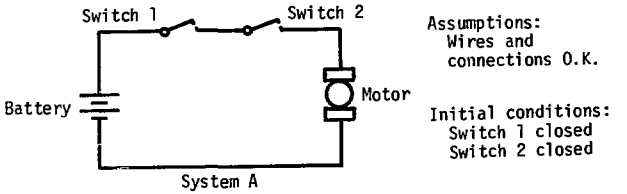


FIG. 1.9 Description of System A

The AND gate describes a causal relationship, the OR gates does not. The input events to an AND gate cause the output event to occur. The output of an OR gate is simply a redefinition of the input.

AND gates can be classified in three categories according to their inputs. In the first class of AND gates, each input is totally independent of the other, i.e., the occurrence of one event has no influence

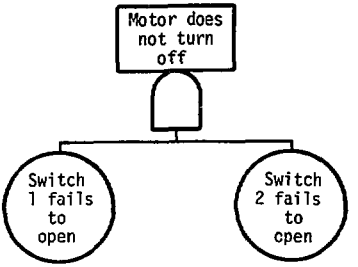


FIG. 1.10 Example of AND Gate Development

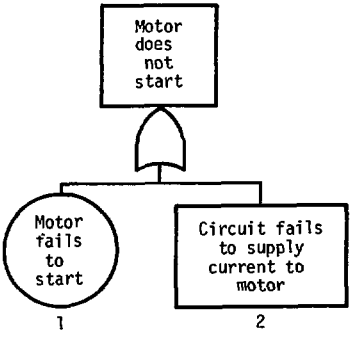


FIG. 1.11 Fault Tree for System A Illustrating OR Gate Development

on the occurrence of the other(s) and vice versa. In the second class of AND gates, called priority AND gates, the one input is dependent on

the occurrence of the other independent input event if the output event is to occur. This dependence, referred to as unilateral, is common for standby and safety systems. Figure 1.12a gives an example of a priority AND gate. Note that the order in which the input events occur is relevant in causing the output event to occur. In the example, should the radiation monitor inadvertently energize the scram magnets after the control rods dropped into core, a successful scram would still have been accomplished and the output event would not have occurred.

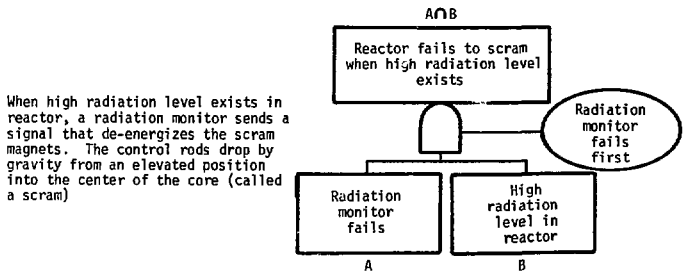


FIG. 1.12a An Example of Priority AND Gate

In the third class of AND gates, the input events are mutually dependent. As an example of mutual dependence, consider two power supplies in parallel feeding a common load. Each power supply can accomodate the entire load but has a higher failure probability when operating alone. The sequence of events that lead to the event "system power failure" is depicted in Fig. 1.12b using one OR and two AND gates with mutually dependent inputs.

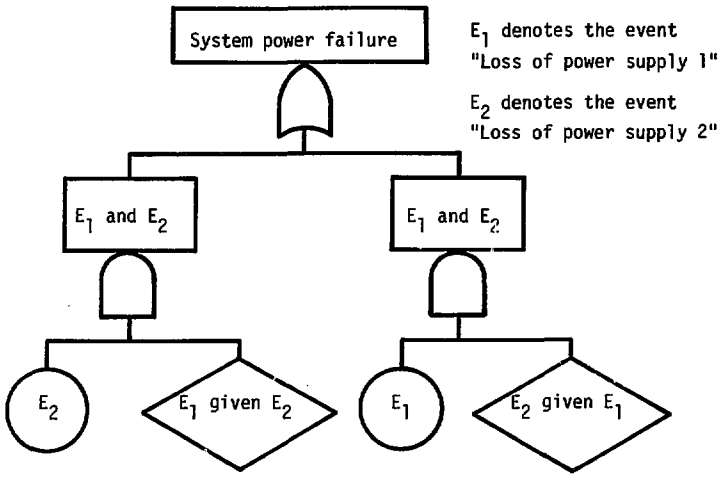


FIG. 1.12b Example of AND Gate With Mutual Dependence

OR gates can also be classified in a similar manner according to their inputs [27]. For the first class of OR gates, the inclusive OR gate, if at least one input event occurs, the output event occurs. The second class of OR gates, exclusive OR gates, the output event occurs if and only if one input occurs, otherwise the output event does not occur. The third class of OR gates, the mutually exclusive OR gate, the occurrence of one input event precludes the existence of all other input events which implies that the output event occurs as a result of only one input event.

The probabilistic evaluation of the three classes of AND and OR gates is discussed in Section 2.7.2. It is shown there that priority AND gates do not obey the laws of conditional probability.

The inhibit gate is essentially a one-input AND gate that describes a causal relationship between one fault and another. The inhibit gate defines a situation where the coexistence of an input event and a conditional event is necessary for the output event to occur. It is a special modification of an AND gate and is used primarily for convenience.

The conditional input defines a state that permits the fault sequence to occur and may be either normal to the system or result from failures. The inhibit gate is used to describe out-of-tolerance failure modes of system elements, i.e., secondary failures. As shown in Figure 1.13, the conditional event describes a sensitivity condition for the system element to fail in the mode specified due to some situation or condition. See Figure 1.14 for a specific example.

1.9.2.5 Construction Methodology - As seen in Figure 1.15, the fault tree is so structured that the sequences of events that lead to the undesired event are shown below the top event and are logically related to the undesired event by OR and AND gates. The input events to each logic gate that are also outputs of other logic gates at a lower level are shown as rectangles. These events are developed further until the sequences of events lead to basic causes of interest, called "basic events". The basic events appear as circles and diamonds on the bottom of the fault tree and represent the limit of resolution of the fault tree.

Inhibit gates are used to develop secondary failures, i.e. out-of-tolerance failures. In this case the condition represents a sensitivity condition.

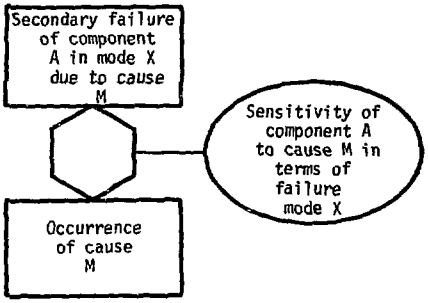


FIG. 1.13 Example of Secondary Failure Development Using Inhibit Gates

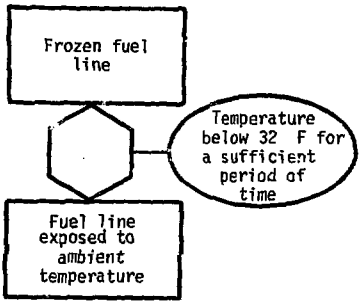


FIG. 1.14 Example of Secondary Failure Development

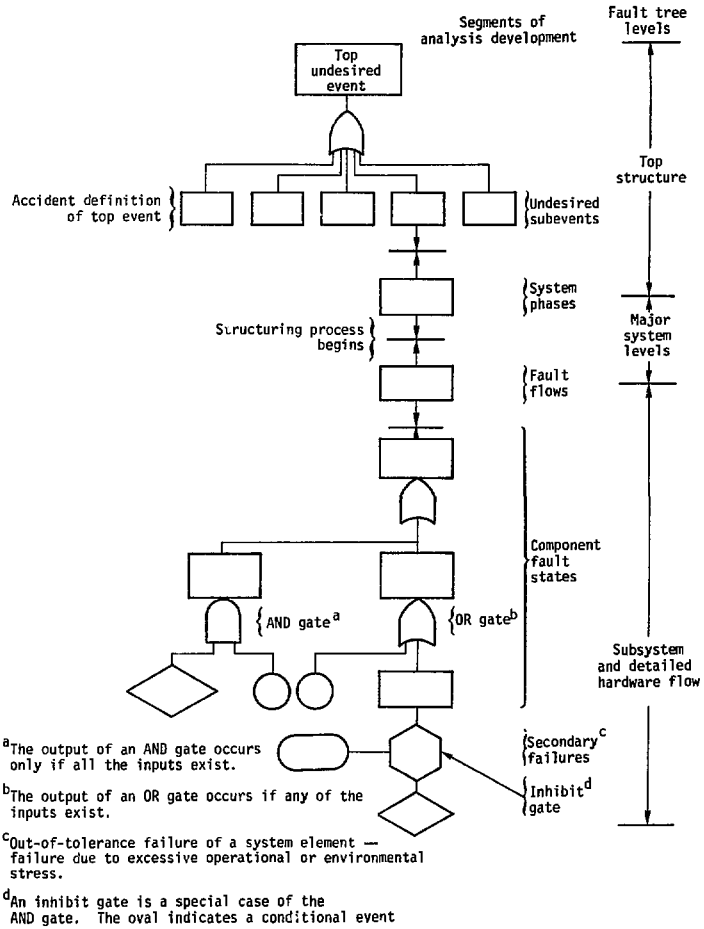


FIG. 1.15 Levels of Fault Tree Development

1.9.2.6 Structuring Process - David Haasl formalized the thought process involved in the construction of the fault tree.* He devised a "structuring process" that established rules to determine the type of gate to use and inputs to the gate. The structuring process is used to develop fault flows in a fault tree (see Figure 1.15) when a system is examined on a functional basis, i.e., when failures of system elements are considered. At this level, schematics, piping diagrams, process flow sheets, etc., are examined for cause-and-effect types of relationships, to determine the subsystem and component fault states that can contribute to the occurrence of the undesired event. At this point, the flow of energy through the system is followed in a reverse sense from some undesirable outcome to its source.

The structuring process requires that each fault event be written to include the description and timing of the fault event at some particular time. This means that each fault event must be written to include what the fault state of that system or component is and when that system is in the fault state. The established procedure answers two principal questions: (1) Is the event a state-of-component or state-of-system fault? (2) what is immediately necessary and sufficient to cause the event?

In a state-of-component fault event, three failure mechanisms or causes are identified that can contribute to a component being in a faulted state.

---

*Much of the material presented in this section on the theory of manual fault tree construction is taken from the course, "System Safety Analysis", given by David F. Haasl et al in the spring of 1972 at Lawrence Livermore Laboratory, Livermore, Califo nia.

1. A primary failure is due to the internal characteristics of the system element under consideration.
2. A secondary failure is due to excessive environmental or operational stress placed on the system element.
3. A command fault is an inadvertent operation or nonoperation of a system element due to failure(s) of initiating element(s) to respond as intended to system conditions.*

The above failure mechanisms describe the fundamental processes involved in or responsible for a component failure mode.

We see that in the case of the first two failure mechanisms, the system element is no longer able to perform its intended function (unless the element is repaired). In the case of the third failure mechanism, the system element can operate as intended if the initiating element(s) is (are) returned to their normal state(s).

We use Figure 1.16 to demonstrate these failure-mechanism concepts. The primary event is indicated in the circle. The command fault is shown in the rectangle. Some out-of-tolerance failure mechanisms for the motor are (1) inadequate maintenance of motor and (2) excessive temperature or external vibration. The fault tree in Figure 1.11 can then be expanded to the fault tree shown in Figure 1.16 to show the development of all three failure mechanisms.

Any fault event that can be described in terms of the failure mechanisms described below is said to be a state-of-component fault event.

---

*An initiating element is any component, human or environmental factor (generally upstream of the element) that can control or limit the flow of energy through the system element under consideration.

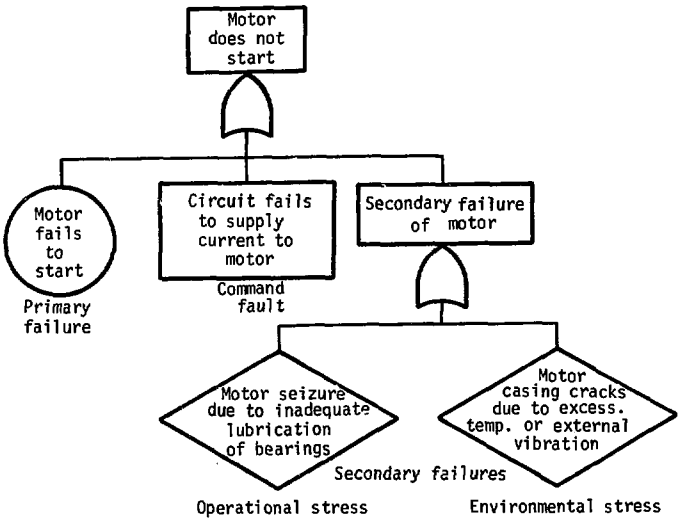


FIG. 1.16 Fault Tree Showing Development of State-of-Component Fault Event

In this case, the system element under examination is the sole cause of the fault event, *i.e.*, the event results from the action of a single component.

An OR gate is always used to combine the inputs at a lower level which consist of the three failure mechanisms or causes as described above. Examples of state-of-component fault events are, (1) failure of motor to start, (2) failure of motor to turn off, (3) switch fails to open, and (4) switch fails to close. Events that have a more basic

cause that cannot be described in terms of a simple component failure are termed state-of-system fault events. In this case an OR gate, AND gate, inhibit gate, or no gate at all can be used to combine the event(s) at the next lower level. In state-of-system fault events, the immediately necessary and sufficient fault input events must be specified. For each newly developed event other than primary causes the structuring process is repeated until each event is developed to its limit of resolution.

To illustrate further the concepts of the structuring process, a detailed fault tree is given in Figure 1.17b for system B as shown in Figure 1.17a. It represents essentially an expansion of the fault tree shown in Figure 1.16. The system description and analysis assumptions that apply to Figure 1.17a are given below.

1.9.2.7 Illustration of Fault Tree Construction System B - System B is a standby system that is tested once every month. It consists of a battery, two switches in parallel, and a motor. To start the motor, two push buttons are pressed to close the two switch contacts 1 and 2. To stop the motor at the end of test, two push buttons are depressed. Periodically, say every six months, the operator must recharge the battery and perform routine maintenance on the motor.

Analysis Assumptions

We assume that the wires or connections do not contribute to system failure. Pre-existing faults are allowed, e.g., the switch contacts may be failed closed as initial conditions. We also assume that all components are properly installed.*

*It is interesting to note that component failures due to improper installation are secondary failures.

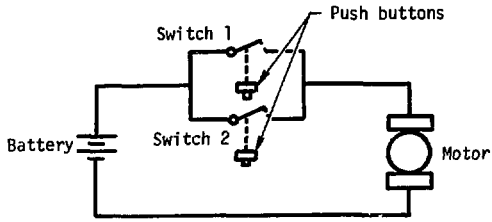


FIG. 1.17a System B

Fault Tree

With reference to Figure 1.17b, the top event appears as failure of motor to start on test implying failure of the motor to start when tested at its monthly interval. Each gate event is labeled as to the event type, either state-of-component or state-of-system fault event. We see that all command faults and secondary failures when developed are state-of-system fault events. An inhibit gate is shown in the development of the secondary failure, c errun of battery. It is interesting to note that two types of failure are shown for the switches. Switches 1 and 2 can fail to close upon demand or they can fail to open from the previous test and cause the battery to discharge. We see that System A and System B are susceptible to one type of failure or the other. A two-out-of-three switch arrangement might be an acceptable alternative. Close examination of the Fig. 1.17b fault tree shows that human error can play a key role in system failure. The operator can forget to

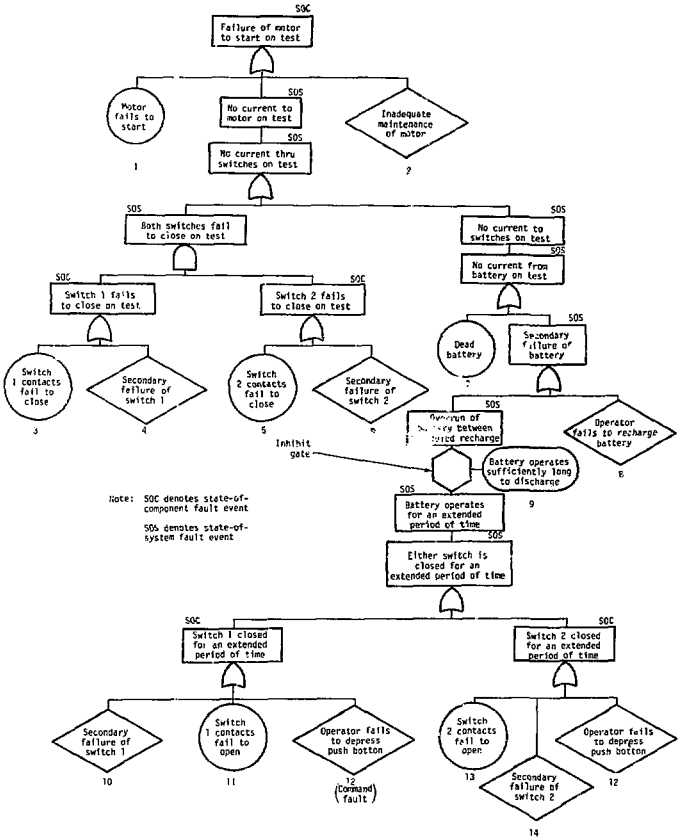


FIG. 1.17b Detailed Fault Tree of System B Generated via Structuring Process

recharge the battery or fail to depress the push buttons after test.* Fault trees that include only hardware failures will overpredict the capability of performance of the system. Realistic assessments of system failure must include human error and secondary failures.

1.9.3 Levels of Fault Tree Development - A complete or global safety analysis using the fault tree technique on an extensive system such as a nuclear power plant or chemical processing plant normally requires three levels of fault tree development as shown in Figure 1.15. The upper structure, called the "top structure," includes the top event and the undesired subevents. These events such as fire, explosion, release of radioactivity are potential accidents and hazardous conditions and are immediate causes of the top event. There is no structuring process at this level to tell the analyst what gate to use or what inputs are specified. The top structure is actually a list of the functions whose loss constitutes a major accident as specified by the top undesired event. David Haasl claims that structuring the fault tree at the top level is an art in outlining. In connection with a recent Air Force contract [39], he made the following statement concerning the content of the upper structure of a fault tree:

> "This level has been defined as the level of clarification and selection. It is at this stage that the comprehensiveness and thoroughness of the planned analysis is determined. This is accomplished by establishing the bounds, both physical and temporal, of the system and determining the limit of resolution of the analysis. In determining the bounds of the system, the effect on the system from inputs

*Note that we are assuming that if the operator fails to depress one push button, he will fail to depress the other push button, i.e., he will skip the procedure entirely. The Reactor Safety Study made similar types of assumptions involving human error.

> from inside the system boundary is considered, but the cause of this effect is not pursued or identified. For the purpose of quantification, it is assumed that any inputs from outside the system boundary are known constants. In determining the limit of resolution, it is assumed either that any finer resolution does not change the effect on the system, or that this effect is a known constant."

The next level of the fault tree divides the operation of the system into phases and subphases, until the system environment remains constant and the system characteristics do not change the fault environment. In this second level of fault tree development, the analyst examines system elements from a functional point of view. Hence, the structuring process is used to develop fault flows within the system that deductively lead to subsystem and detailed hardware fault flow, which is the third level of the fault tree. At the third level, the analyst is faced with one of the most difficult aspects of fault tree analysis. He must show any external failure mechanisms that can simultaneously fail two or more system elements, and restructure the fault tree accordingly. The effects of common environmental or operational stresses are studied, as well as the effects of the human factor in the testing, manufacturing, maintenance, and operation of the system. Some of these factors were considered in the Figure 1.17b fault tree.

1.9.4 Automated Fault Tree Construction - Detailed fault trees of complex systems may take years of effort to complete. Such an effort is generally a costly undertaking. Also, there is a tendency for analysts to become bored constructing fault trees that are large and repetitious. In the process, the analyst may overlook some subtle aspect of system behavior. Therefore, there is a definite need for automated fault tree

construction. It can serve as a tool in assisting an analyst when an in depth safety analysis is required.

In the last five years, efforts have been directed toward automating fault tree construction for computer implementation. Fussell [29] automated fault tree construction for electrical systems. He recognized that there are essentially three ways electrical circuits can fail, (1) no current in circuit when needed, (2) inadvertent application of current and (3) current overload. Powers et al [58] is in the process of automating fault tree construction for chemical systems. In a chemical processing system the situation is more complex than in electrical circuits. Because of the numerous product and reactant streams and diversity of operation it becomes a complex task to locate all the failure pathways and modes of failures for a chemical processing system.

In the sections that follow, the automated approach of Fussell is presented. The method is called the synthetic tree model, STM, and is limited to construction of fault trees for electrical schematics. It is felt that many concepts of the STM can be applied to more general systems such as hydraulic or pneumatic systems. The author regrets that the details of Power's methodology are not available at the time of this writing.

1.9.4.1 Synthetic Tree Model - Fussell's methodology for fault tree construction is programmed in a computer code called DRAFT that automatically constructs fault trees of electrical schematics to the level of primary hardware failures. The basic building blocks of the methodology are component failure transfer functions. These are mini fault trees for components in a faulted state. The information

contained in them can be derived from a failure mode analysis which is independent of the particular system considered. With proper editing, the fault tree is automatically constructed from the component failure transfer functions. A hierarchical scheme is developed that identifies fault events according to order. The information required as input to the code is (1) a schematic of the electrical system, (2) when applicable, the initial operating state of each component and (3) boundary conditions that can impose restrictions on the top event and events developed within its domain.* The computer then finds the series circuit paths for each component in the schematic, called component coalitions, and identifies the order of each event requiring development. Events are considered up to fourth order. It then imposes new boundary conditions when necessary and then constructs the fault tree accordingly. The flowchart illustrating the methodology of the STM is given in Figure 1.18.

1.9.4.1.1 Event Description - In the STM, there are two parts to the event description, (a) the incident identification and (b) the entity identification. The entity identification is the subject of the fault event and refers either to a component or to a component coalition. The incident identification describes a mode of failure or fault state. For example, consider the situation where current is inadvertently applied to the coil of a relay causing its contacts to close. In the fault statement "relay contacts close inadvertently," the entity identification is "relay contacts", and the incident identification is "close inadvertently".

*The domain for the top event includes all events that result from the subsequent development of the top event.

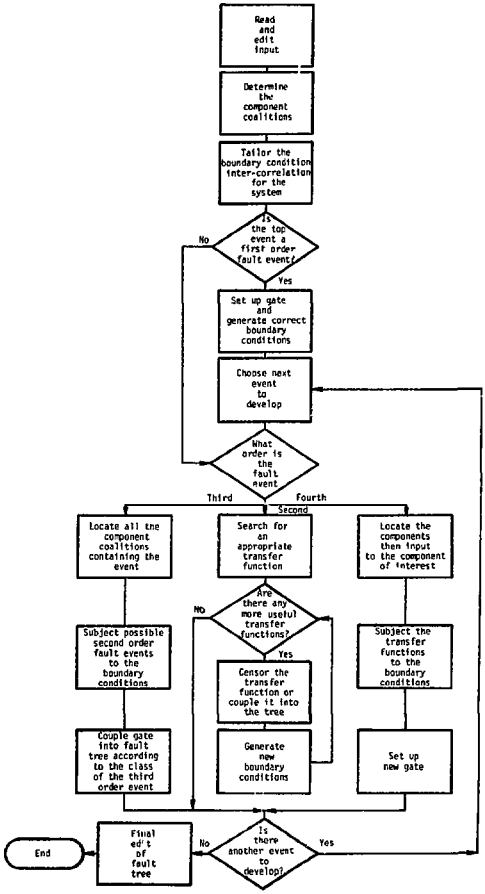


FIG. 1.18 Flowchart for DRAFT Computer Code [29]

1.9.4.1.2 Component Failure Transfer Functions - Primary failures are always part of the component failure transfer functions. The logic gate used in the failure transfer function depends upon the type of failure considered for the component. For example, an electrical component such as a fuse can fail in such a manner as to cause the output event to occur implying OR logic for the output gate. In another case, an electrical component can transmit an overload or inadvertently transmit current. Coexistence of another fault event is necessary for the output event to occur. In this case, the logic for the output gate is AND. This situation is common with protective devices that fail in such a manner to allow out-of-tolerance conditions to exist, e.g., a fuse failing to open when a current overload exists within the circuit. Figure 1.19 illustrates failure transfer functions for electrical contacts. We can see that state-of-component fault events are embodied within these transfer functions.

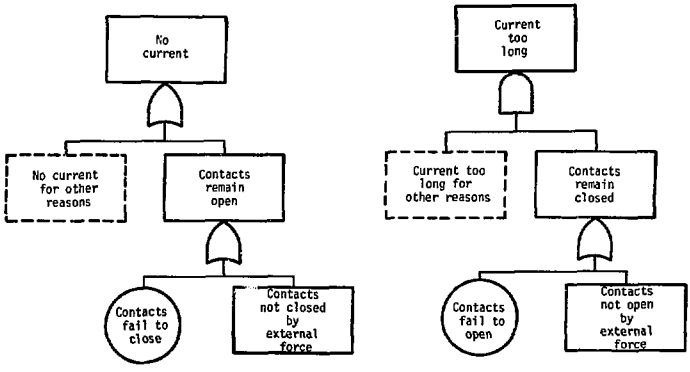


FIG. 1.19 Failure Transfer Functions for Electrical Contacts

1.9.4.1.3 Component Coalition Scheme - Within the context of the STM, a component coalition is a series circuit path in which components share an alliance with respect to current flow. In system B, Fig. 1.17a, there are two component coalitions, (1) the battery, switch 1 and the motor and (2) the battery, switch 2 and the motor. This means there are two paths by which the motor can receive current from the battery.

1.9.4.1.4 Ordering of Fault Events - In contrast to Haasl's structuring process in which there are two basic fault events, state-of-component and state-of-system fault events, in the STM there are four basic types of fault events, (1) first-order, (2) second-order, (3) third-order and (4) fourth-order fault events.* The following paragraphs describe the ordering of the fault events in the STM. It is helpful to refer to the flowchart in Figure 1.17a.

Third and fourth-order fault events in the STM are command faults. For the development of third-order fault events, components are examined with respect to energy input from all series circuit paths that contain these components. This amounts to examining the state of each component coalition that is a source of energy or current to a given component. Events such as "component receives no current when needed" and "component receives current inadvertently" are examples of third order fault events. If a component is producing a fault event because of mechanical linkage with another component, such as a relay coil and its associated contacts or a pressure switch and its contacts, then such an event is

*Fussell's ordering of events is not related to the order of the cut sets.

referred to as a fourth-order fault event. Because of direct component interplay, fourth-order fault events always require component failure transfer functions as input events. Events such as "no current in a component coalition," "overcurrent in a component coalition" and "inadvertent flow of current in a component coalition" are second-order fault events.

The development of third-order fault events always requires as input second-order fault events. In examining the fault state of each component coalition, we must examine each component in the coalition. Hence, the development of second-order fault events always requires as input failure transfer functions. If these failure transfer functions require third- or fourth-order fault events as input, then the above process is repeated until there are no more second-, third-, or fourth-order events that require development. The fault tree is complete when all events are developed to the level of primary hardware failures.

In some cases the top event is of first order, i.e., an event that requires development to the level of subsystem functional faults. In this case the analyst must manually construct the fault tree to the level where events are second order or higher. This procedure is analogous to the construction of the upper structure of the fault tree mentioned in the previous section. Fussell calls the upper structure the tree top boundary condition.

As an example of the synthetic tree methodology, we again construct a fault tree for system B in Figure 1.17a. In the STM, the initial conditions must describe the system in an unfailed state. The system boundary conditions are:

TOP EVENT = Motor fails to start on test

Initial Conditions = Switches open

Not-allowed Events = Wiring or connection failures

Existing Conditions = None.

The fault tree is shown in Figure 1.20. Note that a little more detail is shown on the switch contacts in Figure 1.20 in order to illustrate the development of fourth-order fault events. The hierarchical scheme illustrating the ordering of fault events is evident in Figure 1.20. Also, note that the circled events bear almost exact resemblance to the component failure transfer function given in Figure 1.19 with initial conditions, "contacts open".

Second-order fault events such as "no current in component coalition" impose restrictions on events placed in their domain; e.g., if in the subsequent development of this event, we consider the component coalition again, events like "current in component coalition" are not allowed. Because of this restriction, component failure transfer functions with output event "current" are equally not allowed. Fussell calls these restrictions, event boundary conditions. Such conditions are of consequence when we try to develop the secondary failure "overrun of battery". As we see in the system B fault tree, Figure 1.17b, that the battery discharges when the motor operates for an extended period of time. This further implies there is current in either component coalition 1 or 2. In the context of the STM we cannot place the secondary failure of the battery in the domain of the second-order fault events given in Figure 1.20. Instead, we must consider the system in a different operating state and construct a new fault tree with different tree top boundary conditions. The boundary conditions in this case are:

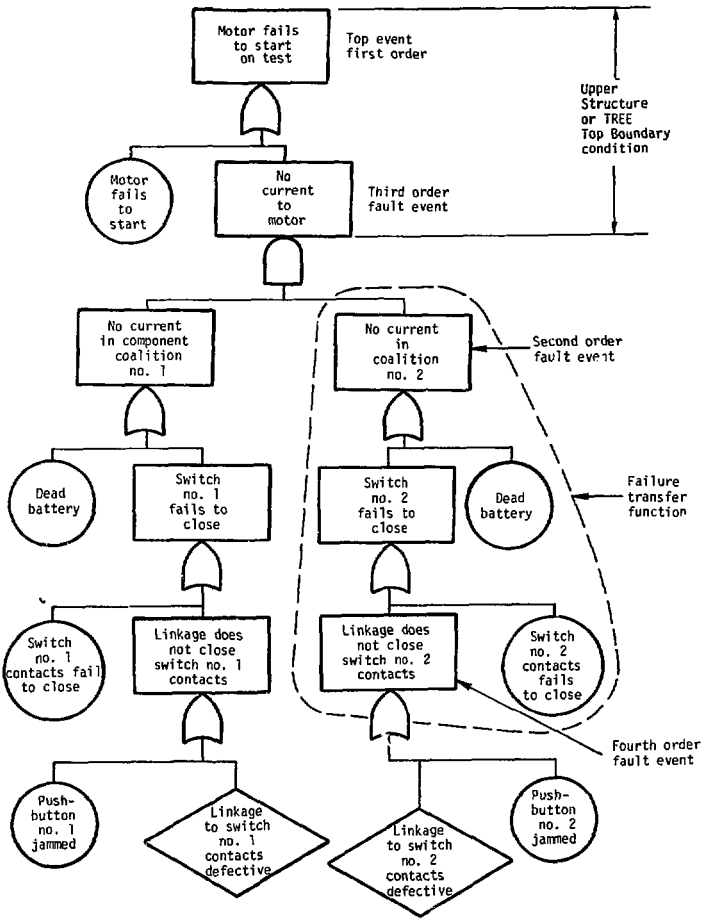


FIG. 1.20 Fault Tree of System B Illustrating Synthetic Tree Methodology

TOP EVENT = Battery operates for extended period of time

Initial Conditions = Switches closed

Not-allowed Events = Wiring or connection failures

Existing Conditions = Motor operating.

The tree top boundary condition and the fault tree are given in Figure 1.21.

Fussell further assigns third order fault events to classes. In Figure 1.21, a component (in this case, the motor) can inadvertently receive current (or an overload) from any coalition containing the component, implying OR logic as shown. This type of third order event is assigned to class I. On the other hand, in Figure 1.20, a component receives no current when needed if all coalitions containing the component have no current, implying AND logic as shown. This type of third-order fault event is assigned to class II. In the DRAFT computer code, identification of the class of third-order events is necessary for determination of the proper logic gate to use, see Figure 1.18.

We see for system B in Figure 1.17a, if switch 1 or 2 is closed, we would expect the motor to operate. The event "current to switch too long" in Figure 1.21 is an existing condition and can be removed from the fault tree. The AND gate can also be removed; the fault then can simply be cascaded from one event to the other.

1.9.5 Manual Versus Automated Fault Tree Construction - We see that the fault tree of Figure 1.17b which was generated via the structuring process does not explicitly show a component coalition. The logic and the fault events that appear in Figure 1.17b are inferred when the schematic in Figure 1.17a is examined. Automated fault tree

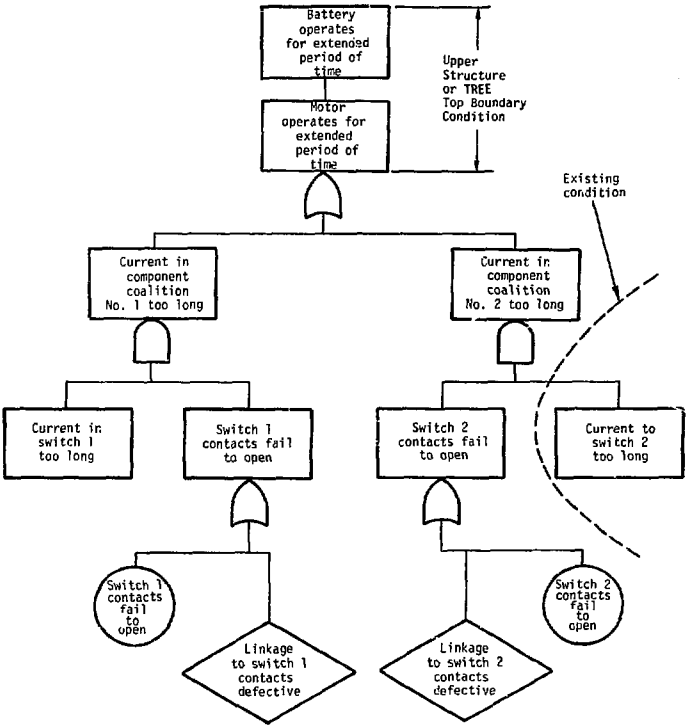


FIG. 1.21 Fault Tree for Secondary Failure of Battery

construction requires that the flow of energy through the system be identified by a method like the component coalition scheme. Also, the computer must generate fault trees from a given set of initial and existing conditions, and events that appear in the fault tree cannot be mutually exclusive. The analyst, on the other hand, can allow various sets of initial and existing conditions to be applicable to the top event. He can logically combine mutually exclusive events through the use of OR gates. This is done in Figure 1.17b for the OR gate with output event "no current through switches on test". For the input event, "both switches fail to close on test," the initial condition that is implied is that the switch contacts are open. What is further implied is that current is available in the circuit at test. For the other input event, "no current to switches on test," the switch contacts can be closed from a previous operation, causing the battery to drain with no current available at test. With large fault trees, the analyst may have a tendency to erroneously combine events that are mutually exclusive through AND gates. In this case, the logic of the fault tree is incorrect because the analyst did not consider the boundary conditions that are applicable to the domain of the AND gate. This problem is discussed further in a latter section (1.9.7) of this chapter.

A disadvantage to the DRAFT computer code is that computer memory storage may be exceeded for large fault trees. This is due to the fact that the computer must store all the event boundary conditions that are generated during the course of fault tree development.

1.9.6 Qualitative Evaluations of Fault Trees - The fault tree can be used as a visual medium in communicating and supporting decisions

based on the analysis. Either the analyst or the management can inspect the fault tree and determine by engineering judgment the most likely sets of basic events leading to the top event. A qualitative judgment can be made regarding the safety of the system and the identification of critical system elements if the system is to be upgraded. A qualitative evaluation can also take into account many practical considerations and assumption that at times may be difficult to incorporate in quantitative calculations. The results of a qualitative evaluation, however, are less manageable due to the subjective nature of decisions based on qualitative judgment.

1.9.6.1 Minimal Cut Sets - The first step in a qualitative evaluation is to determine the minimal cut sets. A minimal cut is a set of basic events whose occurrence causes the top event to occur; it cannot be reduced and still insure occurrence of the top event. For example, a series system of two components, A and B, fails if either A fails or B fails. Considering primary failures only, system failure is defined in terms of two minimal (min) cut sets of one event each: (1) the event "primary failure of A", designated as A, and (2) the event "primary failure of B", designated as B. Note the set of events {A,B} is a cut set but not a minimal cut set. A listing of minimal cut sets is useful for qualitative evaluation. Seventeen minimal cut sets are shown in Table 1.1 for the fault tree of Figure 1.17b. Note that the inhibit condition, "battery operates sufficiently long to discharge" is treated as a basic event and appears in five cut sets, i.e., it is replicated five times.

TABLE 1.1

Listing of Minimal Cut Sets for Fault Tree Given in Figure 1.17b

Cut Set Number	Cut Set	Description
1	{1}	Motor fails to start
2	{2}	Inadequate maintenance of motor
3	{7}	Dead battery (primary failure)
4	{8}	Operator fails to recharge battery
5	{3,5}	Switch 1 contacts fail to close Switch 2 contacts fail to close
6	{3,6}	Switch 1 contacts fail to close Secondary failure of Switch 2
7	{4,5}	Secondary failure of switch 1 Switch 2 contacts fail to close
8	{4,6}	Secondary failure of switch 1 Secondary failure of switch 2
9	{9,10}	Battery operates sufficiently long to discharge Secondary failure of switch 1
10	{9,11}	Battery operates sufficiently long to discharge Switch 1 contacts fail to open
11	{9,12}	Battery operates sufficiently long to discharge Operator fails to depress push button
12	{9,13}	Battery operates sufficiently long to discharge Switch 2 contacts fail to open
13	{9,14}	Battery operates sufficiently long to discharge Secondary failure of switch 2

Two types of primary failures are listed in Table 1.1, human errors and failures of dynamic components. Dynamic components switch or modify energy flows. They must transfer or change state to perform their intended function. Examples include relays, switches, valves and pumps. Another type of component not appearing in Table 1.1 is a quasi-static component. Such components convey or contain energy and include wires, pipes, beams, etc.

In general, human failure rates are one to three orders of magnitude greater than failure rates of dynamic components. In turn, failure rates of dynamic components are one to three orders of magnitude greater than those of quasi-static components. The analyst can mentally factor in these failure rates when determining the critical primary events.

Another factor that must be consider in FTA is the degree to which basic events are replicated in cut sets. For cut sets of a given order,* the top event is structurally more dependent on basic events that are replicated. Another important factor in determining the critical primary events is the order of the cut sets that contain the primary events. When basic events are not replicated, cut sets of lower order are more important than cut sets of higher order when basic event probabilities are equal.

1.9.6.2 Checking Fault Tree Logic Via Cut Sets - The two methods by which fault trees are constructed, the synthetic tree model and the structuring process can lead to seemingly different results. In the Reactor Safety Study the following statement was made about the limitations of fault tree analysis: [77]

*order refers to the number of basic events in the cut sets.

> ". . . there are different ways fault tree logic can be developed. Thus, two different analysts are likely to produce different trees for the same system. Although both trees may be logically correct and produce the same system failure probability, the fact that they appear considerably different can be confusing."

There are ways, however, to check discrepancies in fault trees generated by two different analysts on the same system. It is sufficient to simply inspect the minimal cut sets and note differences. In cut sets of order two or higher where differences appear, the AND gates that combine basic events must be located to check discrepancies in system failure logic.

1.9.6.3 Common-Mode Failure Analysis - It is difficult to design a system in such a manner that the failure rate of the system is below 10^{-5} failures/year because the system will fail in the common mode rather than in combinations of independent individual component failures. Numerous situations can cause the common mode failure to occur -- unrecognized dependence of a control element in the system, human errors in design, operation or maintenance, or unforeseen environmental stresses. Consult references [14], [32] and [73] for a discussion of common mode failures.

In the context of FTA, common-mode failure analysis deals with identifying the mechanisms that are external to the system elements and can cause simultaneous failure of a number of elements or paths. In the context of a command fault, we are concerned with system interface conditions that result in an unrecognized dependence on a control element. This means identification of human as well as hardware functional interdependencies. In the context of secondary failures, we are concerned with unforeseen environmental or operational stresses that can

simultaneously fail two or more system elements. The checklists that we generate as in Figure 1.1 for system energy sources and environmental factors can serve as the first source of information in identifying secondary failure mechanisms.

At least two computer codes exist at Lawrence Livermore Laboratory [54] and at Aerojet Nuclear Corporation [22] that qualitatively can account for common dependencies among cut sets by coding basic events according to an alphanumeric designator. The basic events can be coded according to indices that indicate the following dependencies, (1) location, (2) common function, (3) common environment, (4) common design and manufacturing processes and (5) common operation, test or maintenance procedures involving human intervention. The computer can scan the cut sets for the indicated dependencies to assess the potential for common-mode failures.

1.9.7 Modeling Fault Trees According to System Conditions - A common pitfall of fault tree construction is the inclusion of mutually exclusive events within the domain of an AND gate. In this case, erroneous cut sets can be generated that contain mutually exclusive primary events. If these cut sets are included when the fault tree is quantitatively evaluated, the probability of the top event will be conservatively overestimated (perhaps only slightly). It is important to recognize how logical inconsistencies in fault trees are generated. Basically, it is the result of deficient fault tree modeling techniques when the analyst is not careful in defining the conditions for which the top event is applicable.

An example given by Fussell [27] shows how these erroneous cut sets are obtained. A schematic of a sample system is given in Figure 1.22.

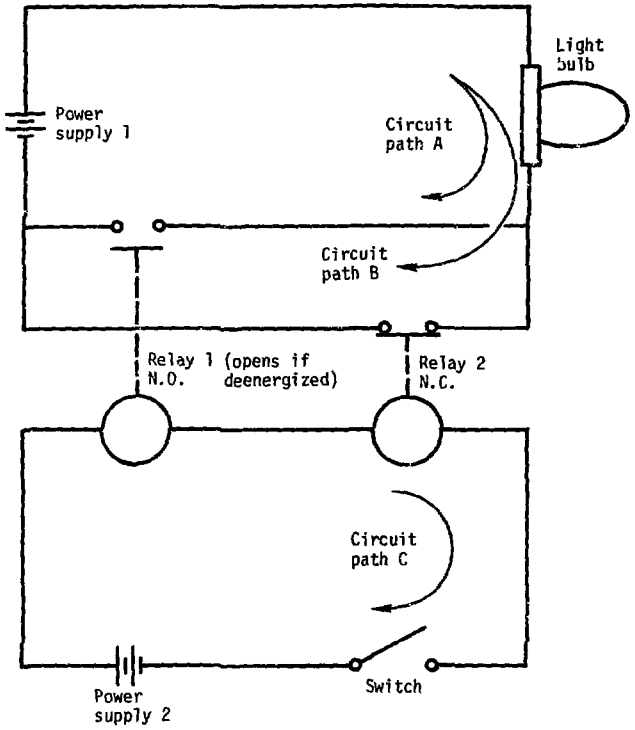


FIG. 1.22 Sample System for Mutually Exclusive Events

The purpose of the system is to provide light from the bulb. When the switch is closed, the relay 1 contacts close and the contacts of the relay 2, (a normally-closed relay) open. Should the relay 1 contacts open the light will go out and the operator will immediately open the switch which in turn causes the relay 2 contacts to close and restore

the light. The system boundary conditions are then:

TOP EVENT	-	No light
Initial Conditions	-	Switch closed
	-	Relay 1 contacts closed
	-	Relay 2 contacts open
Not Allowed Events	-	Operator failures
	-	Wiring failures
	-	Secondary failures.

Operator failures, wiring failures, and secondary failures are neglected to simplify the resulting fault tree. This fault tree generated by conventional techniques is shown in Figure 1.23.

Table 1.2 is a list of minimal cut sets for the fault tree in Figure 1.23.

As Fussell points out, cut sets (6), (8), (10) and (12) will not cause the top event. These cut sets are generated as a result of the logical intersection of two mutually exclusive events "EMF removed from circuit path C" and "EMF not removed from circuit path C". Fussell claims that these events should be flagged so they are never combined to form the erroneous minimal cut sets.

The author claims the fault tree should be modeled correctly in the first place so that mutually exclusive events do not appear in the domain of an AND gate. One should first realize that a fault tree is a static model. Output events of AND gates can exist only under one set of circumstances or (boundary) conditions. At the time that the top event occurs, i.e., when there is no light, either there is current in the lower circuit or there is not, but both situations cannot occur at

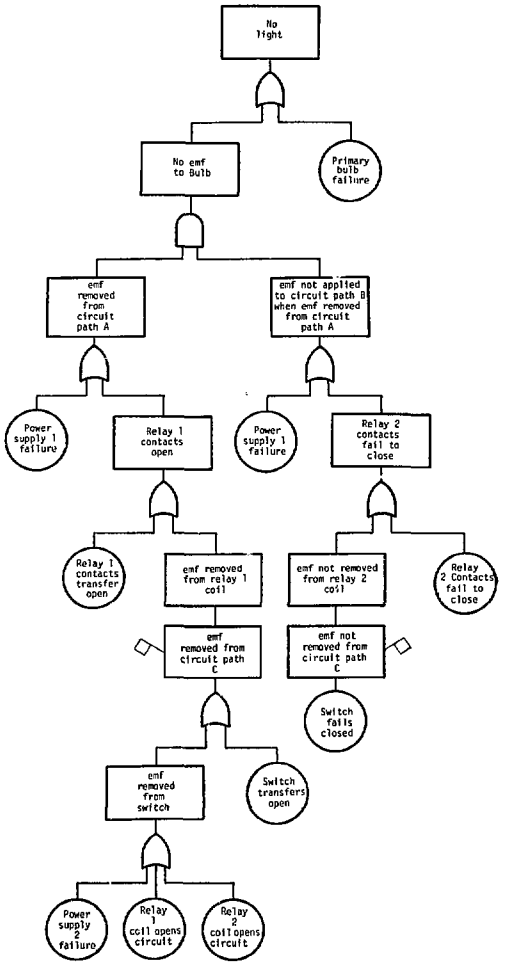


FIG. 1.23 Fault Tree for Sample System in Figure 1.22

TABLE 1-2

Minimal Cut Sets for Sample System

Cut Set Number	Description
1	Primary bulb failure
2	Primary power supply 1 failure
3	Relay 1 contacts transfer open Relay 2 contacts fail to close
4	Relay 1 contacts transfer open Switch fails closed
5	Power supply 2 failure Relay 2 contacts fail to close
6	Power supply 2 failure Switch fails closed
7	Relay coil 1 opens circuit Relay 2 contacts fail to close
8	Relay coil 1 opens circuit Switch fails closed
9	Relay coil 2 opens circuit Relay 2 contacts fail to close
10	Relay 2 coil opens circuit Switch fails closed
11	Switch Transfers open Relay 2 contacts fail to close
12	Switch transfers open Switch fails closed

at the same time. However, both sets of conditions must be considered since they both contribute to the occurrence of the top event. (This situation is analogous to the Figure 1.17b fault tree in which the top event holds for two sets of mutually exclusive system states). The author feels that the "correct" fault tree is given in Figure 1.24. A mutually exclusive OR gate is used. The cut sets in Table 1.3 are the same as given by Fussell except that the conditions under which the cut sets are applicable are explicitly shown.

The erroneous outcome outlined above stems from the tendency of analysts to construct fault trees within the domain of an AND gate that describe fault events sequentially in time according to system operation. For the sample system given in Figure 1.22, however, opening the switch changes the system operating characteristics. It changes the state of the system from "current" to "nocurrent". Again, it may be said that the top event cannot hold for both sets of circumstances simultaneously. When structuring the fault tree as shown in Figure 1.24, it is important to isolate system phases in such a manner that the normal system operating characteristics do not change the fault environment. Otherwise fault trees with erroneous failure logic can be generated.

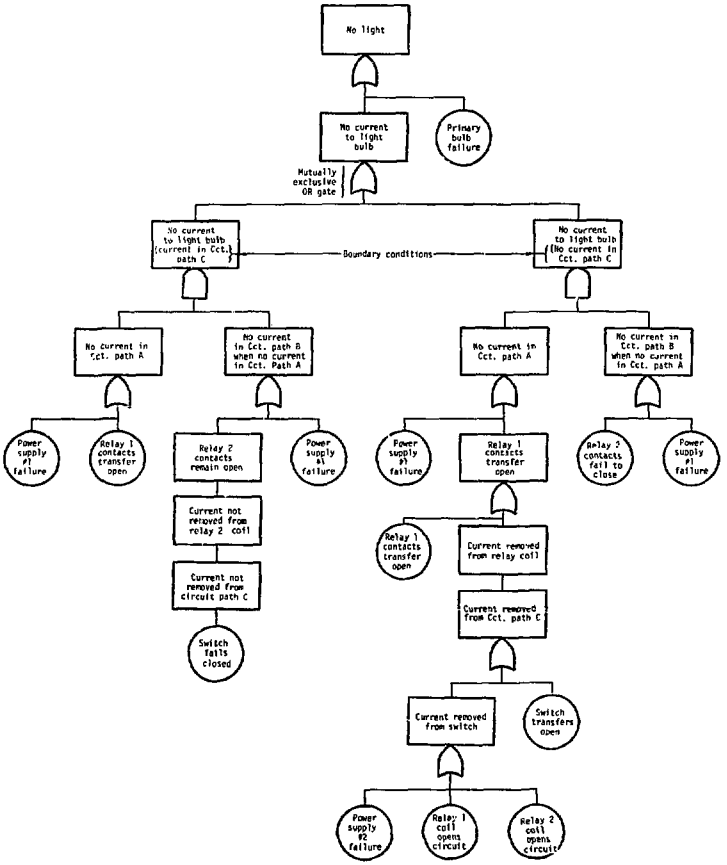


FIG. 1.24 Fault Tree Illustrating Modeling According to Existing Conditions

TABLE 1-3

Minimal Cut Sets for System C

Cut Set Number	Description	Status of System when Light is Off
1	Primary bulb failure	
2	Primary power supply #1 failure	
3	Relay 1 Contacts Transfer Open Switch fails closed	Current in Cct. C
4	Relay 1 Contacts transfer open Relay 2 contacts fail to close	No Current in Cct. C
5	Switch Transfers Open Relay 2 contacts fail to close	No Current in Cct. C
6	Relay 2 coil opens circuit Relay 2 contacts fail to close	No Current in Cct. C
7	Relay ccil 1 opens circuit Relay 2 contacts fail to close	No Current in Cct. C
8	Power Supply #2 Failure Relay 2 Contacts fail to close	No Current in Cct. C

CHAPTER TWO

QUANTITATIVE FAULT TREE ANALYSIS

2.1 Introduction

For newly developed systems in their design stages or for operating systems where failure is rare, we may not have enough information at the systems level to assess with any statistical confidence the probability of system failure. However, if we have failure data at the subsystem or component level then fault tree analysis may be adequate in predicting the probability of system failure as defined by the top event provided the following restrictions are met:

1. The failure data for the basic events are known with sufficient accuracy. (adequacy of data)
2. The fault tree includes all significant system failure modes. (issue of completeness)
3. All failures given in the fault tree can be adequately described in terms of Boolean logic. (binary nature of fault tree modeling)

Chapter Two introduces the reader to the background material necessary for the probabilistic evaluation of fault trees in the context of coherent structure theory [6]. It also describes the role of fault tree analysis in risk assessments by discussing the reliability quantification techniques used in the Reactor Safety Study.

New methods are proposed for (1) determining the unavailability of components due to secondary failures and (2) for finding an upper bound to the distribution of time to first failure and a lower bound on the mean time to first failure for a maintained system.

The importance of min cut sets and basic events can also be computed in terms of mathematical expressions presented in this chapter. Determining the importance of basic events and cut sets is useful when we try to identify critical components for purposes of system upgrade and when we generate repair checklists in the case of system breakdown. Concepts of probabilistic importance within the fault tree framework are presented in Chapter Three and applied to the areas of system design, diagnosis and simulation in Chapters Four and Five.

2.2 Steps in Quantitative Fault Tree Evaluation

The first step in the quantitative evaluation of a fault tree is to find the structural representation of the top event in terms of the basic events, as discussed in Section 2.3. Finding the min cut sets is one way of accomplishing this step. If the rate of occurrence and fault duration time for all basic events are known and the statistical dependency* of each basic event is known (or assumed), then the mathematical expectation (i.e., average) or probability of the top event can be determined. Probabilistic evaluation of fault trees is discussed in Sections 2.4 to 2.8.

2.3 Structural Representations of Fault Trees

2.3.1 Boolean Expression - Following well established nomenclature [6] and procedures, let us first examine the system (i.e., the fault tree) at one point in time. Consider a fault tree with n basic events,

*Two events, A and B with probability P(A) and P(B) of occurrence, are statistically independent if P(A and B) = P(A)·P(B). They are totally dependent if P(A and B) = P(A) = P(B).

the ith event having a binary indicator variable y_i, such that

$$y_i = \begin{cases} 1 \text{ when basic event } i \text{ is occurring} \\ 0 \text{ when basic event } i \text{ is not occurring.} \end{cases}$$

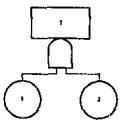
The top event is associated with a binary indicator variable $\Psi(\underline{y})$, such that

$$\Psi(\underline{y}) = \begin{cases} 1 \text{ when the top event is occurring} \\ 0 \text{ when the top event is not occurring} \end{cases}$$

where $\underline{y} = y_1, y_2, \ldots y_n$ is the vector of basic event outcomes. We are assuming that the state of the system $\Psi(\underline{y})$, can be expressed completely in terms of the indicator variables. $\Psi(\underline{y})$ is known as the structure function for the top event.

2.3.2 Logical Operators - There are two logical operators, $\prod$ and $\coprod$, that express Ψ in terms of $\underline{y}$. These are defined and illustrated by examples below.

As an example of the $\prod$ operator, consider the AND gate.



SYSTEM 2-A AND Gate With Two Inputs

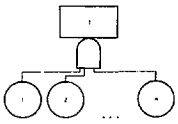
In this case, the top event occurs if basic events 1 and 2 occur. The structure function is given by

$$\Psi(\underline{y}) = \Psi(y_1, y_2) = \prod_{i=1}^{2} y_i \overset{\text{def}}{\equiv} y_1 \cdot y_2.$$

In general, the structure function of an AND gate with n inputs is given by

$$\Psi(\underline{y}) = \Psi(y_1, y_2, \ldots, y_n) = \prod_{i=1}^{n} y_i$$

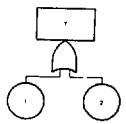
$$\stackrel{\text{def}}{\equiv} y_1 \cdot y_2 \cdots y_n = \min (y_1, y_2, \ldots y_n)$$



AND Gate with n Inputs

The system 2-A fault tree can describe the failure of a parallel system of two components 1 and 2. In this case, the system fails (i.e., the event T occurs) when components 1 and 2 fail (i.e., event 1 and event 2 occur) or $\Psi(1, 1) = 1$ otherwise the system does not fail, i.e., $\Psi(0, 0) = \Psi(1, 0) = \Psi(0, 1) = 0$.

As an example of the $\coprod$ operator, consider the OR gate. In this case, the top event occurs if basic events 1 or 2 occur. The structure function is given by



SYSTEM 2-B, OR Gate with Two Inputs

$$\Psi(\underline{y}) = \Psi(y_1, y_2) = \coprod_{i=1}^{2} y_i \stackrel{\text{def}}{\equiv} 1 - \prod_{i=1}^{2} (1 - y_i)$$

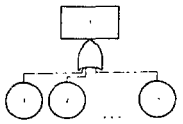
$$= y_1 + y_2 - y_1 \cdot y_2 .*$$

The system 2-B fault tree can describe the failure of a series system of two components. In this case, the system fails when either

*Note that this expression is analogous to the logical union of two events in which $y_1 \cdot y_2$ represents the intersected region on the Venn Diagram.

components 1 or 2 fail, i.e., $\Psi(1, 0) = \Psi(0, 1) = \Psi(1, 1) = 1$. Otherwise the system does not fail, i.e., $\Psi(0, 0) = 0$.

In general, the structure function for an OR gate with n inputs is given by



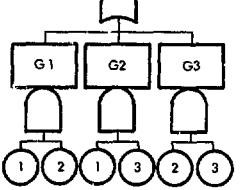
OR Gate with n Inputs

$$\Psi(\underline{y}) = \Psi(y_1, y_2, \ldots, y_n) = \coprod_{i=1}^{n} y_i$$

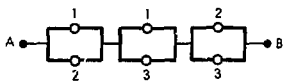
$$\stackrel{\text{def}}{=} 1 - \prod_{i=1}^{n} (1-y_i) = \max (y_1, y_2, \ldots, y_n)$$

Note that $\amalg$ and $\prod$ operate on sets of indicator variables; when pairs of indicator variables are operated on, the symbols π and $\amalg$ are used. By definition, $y_1 \pi y_2 = y_1 \cdot y_2$ and $y_1 \amalg y_2 = y_1 + y_2 - y_1 \cdot y_2$.

2.3.3 Reliability Network Diagram - In general, fault trees are combinations of AND and OR gates. An example of a two-out-of-three system is given below with the corresponding reliability network diagram.



Fault Tree for
2-out-of-3 System



Reliability Network Diagram
for 2-out-of-3 System

The fault tree shown above is "failure oriented"; the numbers in the circles represent component failures. The reliability network diagram is "success oriented". The reliability network diagram can be thought of as an electrical circuit with the circles representing switches. If the components operate successful (switches closed), they transmit the current. The system operates successfully if there is at least one current path from points A to B.

2.3.4 Min Cut Set Representation of $\Psi(\underline{y})$ - As defined in Section 1.9.6.1, a cut set is a set of basic events whose occurrence causes the top event to occur. The terminology "cut set" originated from the reliability network diagram. For example, in the two-out-of-three system, failure of components 1 and 2 constitute a "cut" through the system. For a two-out-of-three system, there are three minimal cut sets {1, 2}, {2, 3} and {1, 3}. In other words, the system fails when at least any two out of three components fail. The structure function is given by

$$\Psi(\underline{y}) = y_1 \cdot y_2 \amalg y_1 \cdot y_3 \amalg y_2 \cdot y_3. \tag{2.1}$$

We must reduce the above expression to its exact Boolean form by expanding the expression to products of indicator variables and then reduce all powers of indicator variables by using the fact that for Boolean variables $y_i^2 = y_i$. The procedure is illustrated below for the two-out-of-three system; successive expansion of expression 2.1 yields:

$$\begin{aligned}\Psi(\underline{y}) &= (y_1 \cdot y_2 + y_1 \cdot y_3 - y_1^2 \cdot y_2 \cdot y_3) \amalg y_2 \cdot y_3 \\ &= y_1 \cdot y_2 + y_1 \cdot y_3 - y_1 \cdot y_2 \cdot y_3 + y_2 \cdot y_3 - y_1 \cdot y_2^2 \cdot y_3 \\ &\quad - y_1 \cdot y_2 \cdot y_3^2 + y_1 \cdot y_2^2 \cdot y_3^2\end{aligned}$$

$$= y_1 \cdot y_2 + y_1 \cdot y_2 - y_1 \cdot y_2 \cdot y_3 + y_2 \cdot y_3 - y_1 \cdot y_2 \cdot y_3$$

$$- y_1 \cdot y_2 \cdot y_3 + y_1 \cdot y_2 \cdot y_3$$

$$= y_1 \cdot y_2 \cdot y_3 + (1 - y_1) \cdot y_2 \cdot y_3 + y_1 \cdot (1 - y_2) \cdot y_3$$

$$+ y_1 \cdot y_2 \cdot (1 - y_3).$$

Expression 2.1 is also known as the min cut representation. In general, the structure function $\Psi(\underline{y})$ may be expressed in terms of the min cut sets as follows

$$\Psi(\underline{y}) = \coprod_{j=1}^{N_k} \kappa_j \text{ where } \kappa_j = \prod_{i \epsilon K_j} y_i$$

where $i \epsilon K_j$ means "for all basic events contained in min cut set K_j"

κ_j = binary indicator variable for cut set K_j

N_K = total number of min cut sets representing the fault tree structure

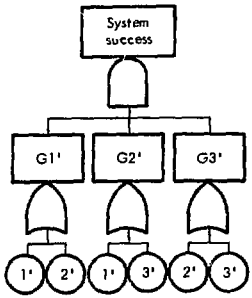
for our two-out-of-three system

$$N_K = 3$$

$$\kappa_1 = y_1 \cdot y_2 \quad \kappa_2 = y_1 \cdot y_3 \quad \kappa_3 = y_2 \cdot y_3.$$

2.3.5 Min Path Representation for $\Psi(\underline{y})$ - In terms of a reliability network diagram, a path set is a set of components whose successful operation insures successful system operation. In the context of fault tree analysis, a min path set is a set of events whose nonoccurrence insures nonoccurrence of the top event. The min path sets are obtained using the duality principle [3]: We change all AND gates to OR gates and all OR gates to AND gates and all events to their complements (indicated by primes). In the case of the two-out-of-three system of Section

2.3.3, the top event "system failure" becomes "system success". The new logic diagram is called the dual fault tree, or success tree, and is shown below.



Dual Fault Tree for 2-out-of-3 System

While the events of the fault tree represent failures, their complements denote successful operation of the components. More generally, complements or dual basic events correspond to the nonoccurrence of the original basic event. The min path sets of the original fault tree are found by obtaining the min cut sets of the dual fault tree. The min path set representation for $\Psi(\underline{y})$ is then given by

$$\Psi(\underline{y}) = \coprod_{r=1}^{N_p} \rho_r \quad \text{where } \rho_r = \coprod_{i \in P_r} y_i$$

where $r \in P_r$ means "for all basic events contained in min path set P_r"

ρ_r = binary indicator variable for min path set P_r

N_p = total number of min path sets representing the fault tree structure.

For the two-out-of-three system

$$\Psi(\underline{y}) = (y_1 \amalg y_2) \cdot (y_1 \amalg y_3) \cdot (y_2 \amalg y_3)$$

$$= (y_1+y_2 - y_1 \cdot y_2) \cdot (y_1+y_3 - y_1 \cdot y_3) \cdot (y_2+y_3 - y_2 \cdot y_3).$$

Expanding as before and reducing powers of indicator variables, we get the same Boolean expression as before.

For complex fault trees, reducing the structure function to its exact Boolean form is an arduous task. When quantifying the fault tree, however, we can obtain useful bounds on the probability of the top event, in terms of the min cut sets and min path sets without a Boolean expansion.

2.3.6 Computer Codes that Produce Cut Sets and Path Sets of Fault Trees · Large fault trees may contain thousands, maybe millions, of min cut sets. Algorithms that find cut sets and are suitable for computer implementation have been devised.

MOCUS [31] is such a computer program based on a deductive algorithm that starts with the top event and generates a two-dimensional matrix. The procedure is equivalent to a series of Boolean expansions of the top event. Each row in the matrix represents the logical intersection of primary and intermediate gate events. The top event is represented by the logical union of all rows in the matrix. The expansion of the matrix is complete when all gate events are expressed in terms of basic events. At this point, each row in the matrix represents a cut set, though not necessarily a min cut set. By the law of absorption, nonminimal cut sets are eliminated.

MICSUP [56] is a computer code based on an inductive algorithm that is an upward Boolean expansion of the fault tree. It starts with the lowest level gates that have basic events as inputs only, finds the min cut sets to these gates and then successively substitutes these cut sets

to these gates. The procedure is repeated until the min cut sets to the top event are found. In general, MICSUP requires less memory storage space in the computer than MOCUS since MICSUP stores all cut sets in a single array.

The SETS computer code [84] finds the "prime implicants" to a fault tree. The prime implicants are like minimal cut sets except that they may contain complemented basic events. The code accepts mutually exclusive OR gates and NOT gates. These gates and complemented events are not accepted in the MICSUP or MOCUS codes.

2.3.7 Coherent Structures - We limit ourselves to Boolean structures, $\Psi(\underline{y})$, that are monotonic or coherent. A coherent structure, $\Psi(\underline{y})$ by definition, is nondecreasing in each argument y_i, i.e., that the occurrence of a basic event cannot cause a system transition from a failed state, $\Psi(\underline{y}) = 1$, to an unfailed state, $\Psi(\underline{y}) = 0$.* This implies that we do not allow complemented events. A coherent structure contains, by definition, all relevant basic events, i.e., the occurrence of each basic event must contribute in some way to the occurrence of the top event. The union of all min cut sets contains all relevant events and is a coherent structural representation for the top event. Formally, $\Psi(\underline{y})$ is coherent if

$$\Psi(\underline{y}) = 1 \qquad \text{if } \underline{y} = (1, 1, \ldots, 1)$$

$$\Psi(\underline{y}) = 0 \qquad \text{if } \underline{y} = (0, 0, \ldots, 0)$$

$$\Psi(\underline{y}) \geq \Psi(x) \qquad \text{if } y_i \geq x_i \text{ for all } i.$$

*This statement has the following engineering interpretation: the degradation of the performance of a system component can only cause the performance of the system to degrade.

Many useful results have been obtained in reliability theory for coherent structures. [5] These are used extensively throughout this thesis.

2.3.8 Structural Dependence and Critical Cut Vectors - Structural dependence is an indication of a functional dependence on basic or intermediate events. A fault tree with n basic events has 2^n possible system states. The number of system states in which the occurrence of event i is critical, known as critical cut vectors, is an indication of structural dependence of the occurrence of the top event. A basic event i is said to be critical for a system state $\underline{y}$ if the system makes the transition from the unfailed state to a failed state when basic event i occurs, i.e., $\Psi(1_i, \underline{y}) - \Psi(0_i, \underline{y}) = 1$.* The vector $(1_i, \underline{y})$ is known as critical cut vector and the set of basic events whose indicator variables equal one in $\underline{y}$ is known as critical cut set for basic event i. The concepts of structural dependence and critical cut vectors are further discussed in Section 3.2.2.1.

2.4 Probabilistic Evaluations of Fault Trees

We have considered thus far the deterministic or structural properties of fault trees. We now consider the probabilistic aspects of FTA.

Again, let us examine the system at one point in time. We assume that the state of the i^{th} basic event is described by a random variable, Y_i. Y_i is a Bernoulli random variable, its probability of occurrence, q_i, is given by the mathematical expectation of Y_i, denoted as $E[Y_i]$,

*The notation $(1_i, \underline{y})$ and $(0_i, \underline{y})$ represents the outcome vectors $(y_1, y_2, \ldots y_{i-1}, 1, y_{i+1}, \ldots)$ and $(y_1, y_2, \ldots y_{i-1}, 0, y_{i+1}, \ldots)$.

where by definition

$$E[Y_i] = 1 \cdot P[Y_i = 1] + 0 \cdot P[Y_i = 0] = P[Y_i = 1] = q_i.$$

Likewise, $\Psi(\underline{y})$ is a Bernoulli random variable, the probability of the top event, P[Top Event] being given by

$$E[\Psi(\underline{Y})] = P[\Psi(\underline{Y}) = 1] = P[\text{Top Event}].$$

If basic events are not replicated in cut sets and all basic events are statistically independent, then

$$P[\text{Top Event}] = \coprod_{j=1}^{N_k} \prod_{i \in K_j} q_i. \tag{2.2}$$

Thus, for statistically independent cut sets and basic events, the expectation "slides" through to each Boolean indicator variable and the structure function is in its exact Boolean form, i.e., there are no powers of indicator variables. In this case, a Boolean expansion is not necessary for calculating the probability of the top event; we merely substitute q_i for Y_i in the structure function.

We can also write

$$P[\text{Top Event}] = \prod_{r=1}^{N_p} \coprod_{i \in P_r} q_i. \tag{2.3}$$

2.4.1 Min Cut and Min Path Bounds to the Probability of the Top Event - In general, basic events are replicated and expressions (2.2) and (2.3) are not valid. Esary and Proschan [18] proved, however, that the following bounds always hold

$$\prod_{r=1}^{N_p} \coprod_{i\epsilon P_r} q_i \leq P[\text{Top Event}] \leq \coprod_{j=1}^{N_k} \prod_{i\epsilon K_j} q_i \qquad (2.4)$$

when the basic events are statistically independent. The upper bound is known as the min cut upper bound and, in general, it is quite close to the "exact" value when the q_i's are small. To illustrate this point, we calculate the upper and lower bounds for the two-out-of-three system. The min path lower bound is given by

$$[q_1 + q_2 - q_1 \cdot q_2] \cdot [q_1 + q_3 - q_1 \cdot q_3] \cdot [q_2 + q_3 - q_2 \cdot q_3], \quad (2.5)$$

and the min cut upper bound by,

$$1 - (1 - q_1 \cdot q_2)\ (1 - q_1 \cdot q_3)\ (1 - q_2 \cdot q_3). \qquad (2.6)$$

Further assume $q_1 = q_2 = q_3 = q$, then expression (2.5) becomes $(2q-q^2)^3$ and expression (2.6) becomes $1 - (1-q^2)^3$.

We plot in Figure 2.1 the upper and lower bounds as a function of q and note that the min cut upper bound is a very accurate approximation. In general, the overprediction that occurs for $.1 \leq q \leq 1$ in Fig. 2.1 is acceptable for most engineering calculations.

The IMPORTANCE computer code discussed in Appendix A accepts as input the minimal cut sets, assumes that all basic events are statistically independent, and conservatively approximates the probability of the top event by the min cut upper bound. The first order expansion of the min cut upper bound is called the rare event approximation. In this approximation we neglect the simultaneous occurrence of two cut sets. As a rule of thumb, the rare-event approximation is accurate when $q_i \lesssim .01$. For example, the first order expansion of expression (2.6) is

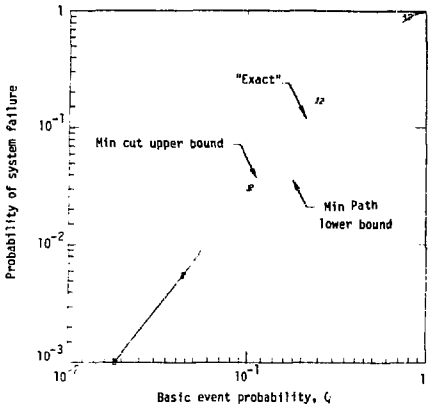


FIG. 2.1 Bounds on System Failure Assuming Independence

given by

$$q_1 q_2 + q_1 q_3 + q_2 q_3 \qquad (2.7)$$

The principle of inclusion-exclusion which is an iterating bounding procedure can be used to find successive upper and lower bounds to the probability of the top event in terms of the min cut sets. Consult reference [6] for a detailed explanation.

2.4.2 Sharper Bounds by Modular Decomposition - Defined in terms of the reliability network diagram, a module is a group of components which behaves as a "super component". In the context of fault trees, an intermediate gate event is a module to the top event if the basic events contained in the domain of this gate event do not appear elsewhere

in the fault tree, i.e., the gate event is a disjoint subtree. Decomposing a tree into modules is useful in reducing the computation required for probabilistic evaluation of fault trees.

A formal definition of a module [6] in terms of coherent structure theory is given as follows: Let Ψ be the indicator variable for the top event depending on a set of basic events N. Let M be a subset of N with complement M^C, χ be a coherent structure on M, then if

$$\Psi(\underline{Y}) = \Gamma(\chi(\underline{Y}^M), \underline{Y}^{M^C}) \tag{2.8}$$

where Y^M means that the arguments are restricted to M, the set M with structure function χ is a module of Ψ. Barlow and Proschan [6] prove under the assumption of statistical independent that the min upper bound is a better (sharper) bound when network diagrams (or fault trees) are decomposed into modules. Chatterjee [10] proposes algorithms to find what he calls the "finest" modular decomposition of a fault tree. Rosenthal [61] has recently written computer codes that modularize fault trees before quantitatively evaluating them.

2.4.3 Computing Bounds When Events are Positively Dependent - The analyst may know that certain components in his system are subjected to a common environment or share a common load, so that a failure of a component, results in increased load on the remaining components. In some cases, it may be difficult or tedious to show this dependency explicitly in terms of a secondary failure development in the fault tree. However, it is possible to incorporate statistical dependency in a quantitative evaluation by assuming that basic events are positively dependent (the

technical term is association).* Esary, Proschan and Walkup [19] show that if indicator random variables are associated, then

$$\max_{1\le s\le k} \prod_{i\in K_s} q_i \le [\text{Top Event}] \le \min_{1\le r\le p} \coprod_{i\in P_r} q_i. \qquad (2.9)$$

Note that in contrast to (2.4), the upper bound here depends on path cut sets.

When basic events are associated, expression (2.9) tells us that the path set with the lowest failure probability is an upper bound for the probability of the top event. For our two-out-of-three system of Section 2.3.3 with $q = q_1 = q_2 = q_3$, expression (2.9) becomes

$$q^2 \le P[\text{Top Event}] \le 1 - (1-q)^2. \qquad (2.10)$$

These bounds are plotted as a function of q in Fig. 2.2, which also shows the probability of the top event assuming statistical independence.

In a series system if we calculate the probability of system failure assuming independence when components are in reality associated, we will overestimate the probability of system failure; in the case of a parallel system, however, we will underestimate the probability of system failure.

The analyst could calculate the probability of the top event by first recognizing independent modules in the fault tree whose basic events are associated. The analyst can then calculate a bound for each module in terms of the path sets as given by expression 2.9. He could then assume that the modules are statistically independent and calculate the probability of the top event in terms of the min cut upper bound given in expression (2.4).

---

*Two random variables X and X are associated if $\text{Cov}[\Gamma(X), \Delta(Y)] \ge 0$ for all increasing binary functions Γ and Δ.

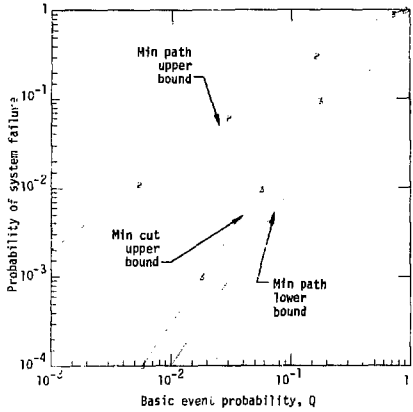


FIG. 2.2 Bounds on System Failure Assuming Association

2.5 Basic Event Characteristics

Initially fault tree analysis was applied to systems that were built and operated remotely such as rocket and satellite systems. These systems were comprised of subsystems and components that were unrepairable during system operation. System success was defined as operating the system without failure for a given mission time. Component failures in this case have an infinite fault duration time. The component failure probability as well as the system failure probability increase as a function of time.

Later fault tree analysis was applied to nuclear power plants and other systems in which repair, inspection and maintenance of system components were an integral part of system operation. In this case,

components have a finite fault duration time. The probability of a component being in a failed state at a certain time, called component unavailability*, approaches an asymptotic limit. The system unavailability in this case is time invariant throughout the life of the system except at its very early stages. There is one distinguishing feature between the two kinds of systems mentioned above. In the former case, where repair is not permitted, components as well as the system can fail only once; in the latter case, where repair of components is permitted, the system can fail more than once.

We now turn to the fundamental probabilistic relations that describe the occurrence of basic events in time.

2.5.1 Basic Events with an Infinite Fault Duration Time - We assume at first that when a basic event occurs, it remains in the ON state for the entire system life.

Let $Y_i(t)$ be a random variable defined as

$$Y_i(t) = \begin{cases} 1 \text{ if basic event } i \text{ occurs (i.e., is ON) by time } t \\ 0 \text{ otherwise} \end{cases}$$

If the occurrence of event i denotes a component failure, then it is customary in FTA to denote

$$E[Y_i(t)] = F_i(t)$$

where $F_i(t)$ is the cumulative failure distribution, i.e., the probability that component i fails over the time interval $[0, t]$. The basic relationships that determine $F_i(t)$ are discussed below.

---

*Unavailability is the probability of a component being in a failed state (being down) at any given time.

2.5.1.1 Life Distribution, Density, Failure Rate - The life distribution of component i, $\bar{F}_i(t)$, is given by

$$\bar{F}_i(t) = 1 - F_i(t).$$

Another fundamental quantity is the failure density, $f_i(t)dt$, defined as the probability that a component fails in a differential time interval, dt about t. If the derivative of $F_i(t)$ exists at t, then

$$f_i(t) = \frac{dF_i(t)}{dt}.$$

A probabilistic function that describes the notion of aging is the failure rate*, $\lambda_i(t)dt$, defined as the probability that component i fails in a differential time interval dt about t given no failure to time t. Hence, $\lambda_i(t)dt$ is a conditional probability and is given by

$$\lambda_i(t) = \frac{f_i(t)}{1-F_i(t)}$$

when $f_i(t)$ exists and $F_i(t) < 1$. The failure rate can be expressed in terms of time units (e.g., hours) or in terms of operating cycles. Integrating the above expression, then exponentiating we get

$$F_i(t) = 1 - e^{-\int_0^t \lambda_i(t')dt'}$$

The cumulative failure rate, $R_i(t) = \int_0^t \lambda_i(t')dt'$, and is referred to as the hazard.

The time dependence of the failure rate of a component, in many cases, is given by the familiar bath tub curve. In their early life, components experience a burn in, or debug period, also known as infant-

*$\lambda_i(t)$ is also known as the hazard rate, force of mortality or intensity rate.

mortality period, in which components experience a high failure rate. Then for a large portion of the component's life, known as the useful life phase, the component experiences a constant failure rate in which failures are random. In the late part of the component's life, known as the wear-out period, the component experiences an increasing failure rate. As shown in Figure 2.3 [42], electrical components generally display a more constant failure in the useful life phase than do mechanical components. Quality control can eliminate most failures due to burn in by testing. A proper maintenance program can insure that most components do not operate in the wear-out region.

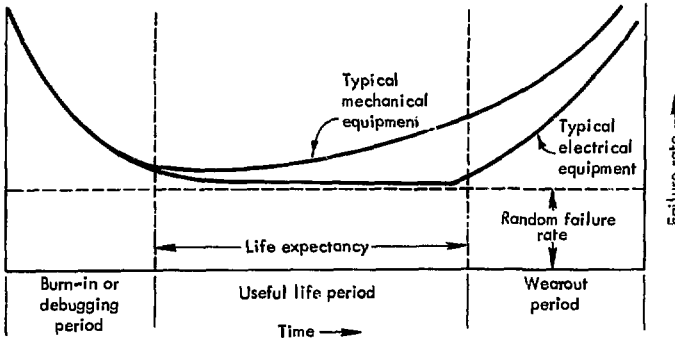


FIG. 2.3 Time Dependence of Failure Rate

Failure rates that are constant in time are characterized by the exponential distribution; the cumulative failure distribution in this case is given by

$$F_i(t) = 1 - e^{-\lambda_i t}$$

where λ_i, the failure rate, is a constant. In this case, the density $f_i(t)$ is given by

$$f_i(t) = \lambda_i e^{-\lambda_i t}.$$

Failure described by the exponential distribution is a memoryless process. Given successful operation at time T, the probability of failure in a given time interval, Δt about T, i.e., [T, T+Δt], is constant and does not depend on T.

Examples of two-parameter life distributions are the Weibull, gamma and log normal distributions. The Weibull distribution is used to describe non-steady state behavior such as burn in or wear out. The gamma distribution is useful for characterizing asymetric one-peak behavior of the density function. The log-normal distribution is useful for describing failures characterized by multiplicative contributions. (See Section 4.1.2). These distributions are discussed in references [3], [6], and [79].

Failure rates may be a function of the environment. For example, Vesely [78] reports that identical components (same manufacturer) but located at two different nuclear power plants had failure rates that varied by two orders of magnitude. Bourne and Green [36] allow for adjustment of failure rates by multiplicative constants, called K factors. These are functions of the component's environmental condition, percentage of nominal rating, and temperature. Subjective judgment is generally required in the assignment of these K factors.

2.5.1.2 Mean Time to Occurrence - Another fundamental quantity is the mean time to occurrence of a basic event, $\bar{m}$, where by definition

$$m = \int_0^\infty t\, f(t)dt$$

integration by parts shows that

$$m = \int_0^\infty \bar{F}(t)dt.$$

If a component has an exponential life distribution, then its mean time to failure, μ, is given by

$$\mu = \int_0^\infty e^{-\lambda t}dt = 1/\lambda.$$

2.5.2 Basic Events with a Finite Fault Duration Time - Basic events that can alternate between the OFF state and the ON state have a finite fault duration time. If we are interested in the time to first occurrence of these events, then the basic probabilistic quantities of the previous section can be used. However, if we are interested in the probability that an event i is in the ON state at a certain time, regardless of the number of times that the basic event has occurred, then we must introduce the concept of ON availability, defined as the fractional amount of time an event is in the ON state. Formally the ON availability for basic event i is defined by the $E[Y_i(t)]$ where $Y_i(t)$ is now a random variable defined by

$$Y_i(t) = \begin{cases} 1 \text{ if basic event i is occurring (i.e., is ON) at time t} \\ 0 \text{ otherwise.} \end{cases}$$

When a basic event describes a component failure, the fraction of the time the component spends in the failed state (i.e., ON state) is denoted as unavailability and the fraction of time in the unfailed state as availability.

There are basically two kinds of component unavailability. First, we consider interval unavailability which is expressed in terms of a given time interval or cycle time. It is computed by taking the ratio of downtime to some cycle time. Interval unavailability is associated with scheduled testing and maintenance. Later in this section we discuss renewal theory. In that context, we are concerned with point unavailability, i.e., the probability that the component is down at some time.

First let us consider the ON availability of normal events.

2.5.2.1 Normal Events - Normal events are events that are expected to occur and are usually represented by houses. Houses are turned on with probability one during their effective duration. It is erroneous, however, to assume that the ON availability of these events is one when calculating the system interval unavailability. For example, in a continously operating system, we remove a battery for test at the end of each day for five minutes. The interval unavailability of the battery, i.e., its fractional downtime, due to normal causes is $\frac{5}{(60)(24)} \cong .35 \times 10^{-2}$ (and not one!), i.e., the battery is removed .35% of the time during system operation.

2.5.2.2 Fault Events, Component Failures, Maintenance Policies - The unavailability of a component in a system is dependent upon factors such as the length of time a component can remain in the failed

state (i.e., detection time) and upon how long it may take to repair the component (i.e., repair time).

In some cases, components can fail without being detected. For example, failure of a component in a redundant system will not cause the system to fail and if not monitored, the component can remain in a failed state until system failure. Another example is a standby system such as the emergency core cooling system, ECCS, at a nuclear power plant. The ECCS can fail prior to demand and be unavailable upon demand. Testing such systems and components can reduce their unavailability (within some limit) as demonstrated in the next section.

2.5.2.2.1 The Effect of Scheduled Maintenance and Testing on Component Unavailability

Component Unavailability - Consider the following maintenance model

a. A component has a failure distribution F(t).
b. It is inspected every T_I units of time.
c. The component failure is detected only when inspected. The probability of uncovering a failure at inspection is unity.
d. The component is renewed to as-good-as-new status at the end of the inspection interval. To inspect the component, it must be removed from service. On the average, it takes τ_r units of time to inspect and replace the component if found failed.

If $\tau_r \ll T_I$ (i.e., inspection and replacement time is much less than the inspection interval) and $\tau_I T_I$ is a small quantity, a second order

expansion of $e^{-\lambda_i T_i}$ shows that the interval unavailability of i, $\bar{A}_i$ reduces to [3]

$$\bar{A}_i = \lambda_i T_I/2.$$

Testing a component too often can actually increase its unavailability (if the component must be removed from service for testing). Jacobs [44] shows that the optimum inspection interval, T_I, that minimizes the component unavailability for a given inspection and replacement period τ_r is

$$T_I = \sqrt{\frac{2\tau_r}{\lambda_i}}$$

if the component has an exponential life distribution and $\lambda_i T_I$ is a small quantity.

Henley [43] reports for the chemical industry that after performing maintenance, the failure rates of components in many instances increased. Incorporating this fact in determination of an optimum maintenance interval (as given above) is difficult because maintenance and testing actions depend upon humans and their effects are not easily quantified. This brings up an interesting point in the nuclear community -- does testing of the engineered safety system at the frequency of once a month (as specified by NRC) enhance the availability of these systems?

For most systems, a cost penalty is associated with system downtime. Also, many systems are series systems, i.e., any component failure causes the system to fail. If these systems fail, it may be cost effective to replace other components that are wearing out while replacing the failed components. This procedure is called opportunistic replacement and is considered by Sethi in his PhD thesis [66].

2.5.3 Renewal Theory - In many systems, we simply replace or repair components instantaneously as they fail. This procedure is referred to as off-schedule maintenance as opposed to the preventative maintenance mentioned in the previous sections. The process of replacing components as they fail generates a renewal process. Consider the process of operating a component until it fails at time $t_1 = T_1$ and is replaced with an identical component (instantaneously) and fails again at $t_2 = T_1 + T_2$ and is replaced -- this replacement process is repeated in time. The sequence of random variables, T_1, T_2, ...T_n forms a renewal process. The probability that the inter-arrival time T_i (the length of the i^{th} operating period) is less than time t' (t' counted from the start of the i^{th}-1 replacement) is defined by the distribution

$$P(T_i < t') = \Phi(t')$$

and its density

$$P(t' < T_i < t' + dt') = \phi(t')dt.$$

When, for a given component, all inter-arrival times have the same distribution, the above process is referred to as an ordinary renewal process. In some cases, T_1 has a different distribution $\Phi_1(t)$, the the process is a modified renewal process.

The following quantities are fundamental to renewal theory:

1. $P(T_1 + T_2 ... + T_n < t)$: probability that the n^{th} replacement (renewal) occurs before t.
2. N(t): the number of renewals in the interval (0, t).

3. $W(t) \equiv E[N(t)]$: the average number of renewals in the interval (0, t) (renewal function), and
4. $w(t) \equiv \frac{dW(t)}{dt}$: renewal density with interpretation: $w(t)dt$ = probability that a renewal occurs in the interval $(t, t + dt)$. $w(t)$ is a probability density.

It is important to note that the above quantities (1 through 4) are calculated in terms of a time scale t that is counted from the beginning of the renewal process. It can be shown that*

$$W(t) = \phi_1(t) + \int_0^t W(t-x)\ \phi(x)dx,$$

and by differentiating, we generate the renewal density

$$w(t) = \frac{dW(t)}{dt} = \phi_1(t) + \int_0^t w(t-x)\phi(x)dx. \qquad (2.11)$$

The above equation has the following physical interpretation: $w(t)dt$ is the probability that a renewal (and in this case a failure) can occur in one of two mutually exclusive ways: (1) a component can fail for the first time in $(t, t + dt)$ (first term on the right hand side) or (2) a renewal took place at $t-x$ and then the component failed again in $(t, t + dt)$, (second term).

In particular, when all inter-arrival times are exponentially distributed, i.e.,

$$\phi_1(t) = \phi(t) = \lambda e^{-\lambda t}.$$

Equation (2.11) can be solved by Laplace transformation to yield

---

*the exact details of this mathematical development can be found in any book on renewal theory, in specific, consult references [3], [6], [12] and [62]. This development follows reference [3].

$$w(t) = \lambda$$

and

$$W(t) = \lambda t$$

which is to be expected since the exponential distribution is a memoryless process.

An asymptotic result holds for any distribution that is *non*lattice (i.e., nonperiodic) [62] and is independent of $\phi_1(t)$ is

$$\lim_{t\to\infty} w(t) = \lim_{t\to\infty} \frac{W(t)}{t} = \frac{1}{m} \tag{2.12}$$

where m is the mean of $\Phi(t)$. For a component, expression (2.12) tells us that the rate of renewal (and hence failure) is $1/\mu$ in the asymptotic steady state.

2.5.3.1 Alternating Renewal Processes - Instead of replacing components with new ones, we consider now the process of repairing components as they fail. Again, we assume that components fail randomly in time. When a component fails, we assume that it is monitored, that repair takes place immediately and is repaired to as good-as-new status. We also assume that the time required for repair is a random variable. The process of repairing a component as it fails in time in the manner described is an alternating renewal process. In particular, the length of the i^{th} replacement period (or cycle), T_i, is the sum of two independent random variables, X_i and Y_i where X_i denotes the amount of time the component is working during the i^{th} renewal cycle and Y_i, the time the component is under repair. In this case, the density of the inter-

arrival time, $\phi(t)$, is given by* [3]

$$\phi(t) = f(t) * g(t) \tag{2.13}$$

where * denotes the convolution of two random variables, f(t) is the failure density for X_i and g(t) is the repair density for Y_i.

2.5.3.1.1 Renewal Density - The renewal density satisfies the following equation

$$w_r(t) = \phi(t) + \int_0^t w_r(t-x)\ \phi(x)dx \tag{2.14}$$

describes an ordinary renewal process, t denotes the time at which a renewal takes place (i.e., the time the component is restored to working order from a failed state). Equation (2.14) has a similar physical interpretation as equation (2.11): $w_r(t)$ is the probability that a renewal takes place in (t, t+dt) in one of two mutually exclusive ways: (1) the first renewal occurs in (t, t+dt) or (2) the first renewal occured at time x and the component is renewed again in (t, t+dt).

2.5.3.1.2 Failure Density - If we count the times at which failure occurs, then we have a modified renewal process. $\phi_1(t)$ is, in this case, the density f(t) and we can generate an expression for

---

*By the convolution theorem

$$\phi(t) = \int_0^t g(t-x)\ f(x)dx = \int_0^t f(t-x)\ g(x)dx = f(t) * g(t).$$

The Laplace Transform of the convolution is simply

$$L[\phi(t)] = L[f(t)]\ L[g(t)] \overset{\text{def}}{=} \tilde{f}(s)\cdot\tilde{g}(s)$$

and makes the calculation for W(t) possible.

the failure density, $w_f(t)$

$$w_f(t) = f(t) + \int_0^t w_f(x)\ \phi(t-x)dx \qquad (2.15)$$

and $w_f(t)dt$ has the following probabilistic interpretation. A component can fail in (t, t+dt) in one of two mutually exclusive ways; it can fail for the first time in (t, t+dt) or it can fail and be repaired (for the first time) at t-x and fail again in (t, t+dt). The expected number of failures in [0, t], $E[N_f(t)]$, is the integral of (2.15) over time, i.e.,

$$E[N_F(t)] = \int_0^t w_f(t)dt.$$

2.5.3.1.3 Availability - By a similar development, we can show [3] that the availability of a component, p(t), for an alternating renewal process is given by

$$p(t) = 1 - F(t) + \int_0^t w_r(x)\ [1 - F(t-x)]dx \qquad (2.16)$$

where F(t) is the failure distribution of f(t), the failure density.

Expression (2.16) has the following physical interpretation, the probability of a component being up at time t is the result of two mutually exclusive events, (1) the component does not fail at all in (0,t) or (2) repair occurs at x and a failure does not occur in [t-x, t].

Usually we have that p(0) = 1. The unavailability q(t) is simply*

$$q(t) = 1 - p(t).$$

---

*Notation: A(t) is equivalent to (≡) p(t) and $\bar{A}(t) \equiv q(t)$.

2.5.3.1.4 Asymptotic Results - Of interest are the asymptotic or steady state results for the alternating renewal process.*

$$A = p_{\infty} = \frac{\mu}{\mu+\tau}$$

$$\overline{A} = q_{\infty} = \frac{\tau}{\mu+\tau}$$

$$w_{f,\infty} = w_{r,\infty} = \frac{1}{\mu+\tau}$$

where μ, the mean time to failure, is the mean of F and τ, the mean time to repair is the mean of G. The above results tell us that the rate of renewal and rate of failure in the steady state is $\frac{1}{\mu+\tau}$. Further, with probability one [62]

$$\lim_{t\to\infty} \frac{t}{N_r(t)} = \mu+\tau.$$

The quantity $\mu+\tau$ is the average length of time for a renewal cycle in the steady state. $N_r(t)$ is the number of renewals by time t.

2.5.3.1.5 Case of Exponential Repair and Exponential Failure - In this case, $f(t) = \lambda e^{-\lambda t}$ and $g(t) = \nu e^{-\nu t}$ (note ν is equal to $\frac{1}{\tau}$). Assuming the component is working at t=0, i.e., $p(0) = 0$, simple calculations involving Laplace transforms yield [3]

$$p(t) = \frac{\nu}{\nu+\lambda} + \frac{\lambda}{\nu+\lambda} e^{-(\lambda+\nu)t} \tag{2.17}$$

$$w_f(t) = \frac{\lambda\nu}{\lambda+\nu} + \frac{\lambda^2}{\lambda+\nu} e^{-(\lambda+\nu)t} \tag{2.18}$$

$$w_r(t) = \frac{\lambda\nu}{\lambda+\nu} - \frac{\lambda\nu}{\lambda+\nu} e^{-(\lambda+\nu)t}. \tag{2.19}$$

*The method of obtaining these asymptotic results in shown in Section 2.7.3 when an expression for component unavailability due to secondary-failure causes is derived.

The IMPORTANCE computer code presented in Appendix A assumes that the failure distribution and repair distribution for basic events are exponential and calculate the unavailability and failure density for basic events in terms of the expressions given above assuming that $p(0) = 1 - q(0) = 1$.

2.6 Top Event or System Characteristics

We are now interested in the probability of the top event in the general case in which basic events have either a finite or infinite fault duration time. In the nonrepairable case, it is clear that an occurrence of a basic event at time t is equivalent to occurrence in an interval of time [0, t]. Let us define the basic event indicator variables as

$$Y_i(t) = \begin{cases} 1 \text{ if basic event } i \text{ is ON at time } t \\ 0 \text{ if basic event } i \text{ is OFF at time } t \end{cases}$$

and if $Y_i(t)$ is random, define $E[Y_i(t)]$ as

$$E[Y_i(t)] \overset{\text{def}}{=} q_i(t) = \begin{cases} F_i(t) & \text{if basic event } i \text{ has an infinite fault duration time} \\ \bar{A}_i(t) & \text{if basic event has a finite fault duration time (its ON availability).} \end{cases}$$

If indicator variables are independent, then the system unavailability (the ON availability of the top event) is given by

$$E[\Psi(\underline{Y}(t))] \overset{\text{def}}{=} g\,(\underline{q}(t)) = g\,(\underline{F}(t), \underline{\bar{A}}(t))$$

where $\Psi(\underline{Y}(t))$ is the structure function for the top event and is assumed to be coherent.

In the above definitions, we assume that all repair processes are independent. This implies that each system component is assigned separate repairmen. Calculating system unavailability, for example, when

there is only one repairman for more than one component must be handled by Markov processes.

A fundamental probabilistic quantity, which is going to be used extensively in Chapter Three and beyond is the probability that the system is in a state such that the occurrence of event i is critical. This quantity is given by

$$E[\Psi(1_i, \underline{Y}(t)) - \Psi(0_i, \underline{Y}(t))].$$

Since $g(\underline{q}(t))$ is linear in $q_i(t)$ (since $\Psi(\underline{Y}(t))$ is linear in $Y_i(t)$)

$$\frac{\partial g(\underline{q}(t))}{\partial q_i(t)} = E[\Psi(1_i, \underline{Y}(t)) - \Psi(0_i, \underline{Y}(t))]$$

$$= g(1_i, \underline{q}(t)) - g(0_i, \underline{q}(t))$$

when basic event indicators are statistically independent.

2.6.1 Expected Number of System Failures - We introduce the following notation

$$w_{f,i}(t) = \begin{cases} f_i(t) & \text{(the failure density) if basic event i has an infinite fault duration time} \\ w_f(t) & \text{(the failure density in renewal theory) if basic event i has a finite fault duration time} \end{cases}$$

and define $w_{f,s}(t)dt$ as the probability that the system fails (i.e., top event occurs) in [t, t+dt], i.e., the system failure density.

If it is assumed that only one basic event can fail in a differential time interval, dt, i.e., the probability of two or more events failing in dt is second order or higher. In this case, Murchland [51] showed that for coherent structures

$$w_{f,s}(t) = \sum_{i=1}^{n} \frac{\partial g(\underline{q}(t))}{\partial q_i(t)} \quad w_{f,i}(t). \tag{2.20}$$

The above result is reasonable on physical grounds. If basic events are independent, the top event must be caused by a basic event occurring at some instant of time. The probability that a basic event i causes system failure in dt is then the product of two independent terms; the probability that the system is in a state in which the occurrence of event i is critical and the probability that event i actually fails in [t, t+dt]. The expected number of system failures in [0, t] is

$$E[N_s(t)] = \int_0^t w_{f,s}(t)dt.$$

The expected number of system failures caused by event i in [0, t] is

$$E[N_{s,i}(t)] = \int_0^t \omega_{f,i}(t)dt$$

where by definition, the rate that event i causes system failure at time t is given by

$$\omega_{f,i}(t) \overset{\text{def}}{=} \frac{\partial g(\underline{q}(t))}{\partial q_i(t)} w_{f,i}(t). \tag{2.21}$$

A very interesting result proved by Murchland [51] is that the system unreliability, $F_s(t)$ (one minus the probability of no system failures in [0, t]) is bounded as follows

$$g(\underline{q}(t)) \leq F_s(t) \leq E[N_s(t)]. \tag{2.22}$$

Furthermore, $E[N_s(t)]$ is very close to $F_s(t)$ for small t. The IMPORTANCE computer code written for this thesis and described in Appendix A

computes the upper bound, i.e., the expected number of system failures, and the lower bound, the system unavailability, as a function of time assuming exponential failure and repair rates.

Barlow and Proschan [4] show that if repair is not allowed, then

$$F_s(t) = \sum_{i=1}^{n} \int_0^t (g(1_i, \underline{F}(t)) - g(0_i, \underline{F}(t)))f_i(t)dt.$$

We can calculate the expected number of system failures in terms of minimal cut set failure densities. [81] Define the unavailability of the j^{th} cut set as

$$Q_{K_j}(t) = \prod_{i \in K_j} q_i(t) \tag{2.23}$$

where $q_i(t)$ is the basic event ON availability as defined previously. An expression similar to 2.20 can be used to calculate $w_{f,s}(t)$

$$w_{f,s}(t) = \sum_{j=1}^{N_k} \frac{\partial g(\underline{q}(t))}{\partial Q_{K_j}(t)} w_{f,K_j}(t) \tag{2.24}$$

where the cut set failure density is given by

$$w_{f,K_j}(t) = \sum_{i \in K_j} \frac{\partial Q_{K_j}(t)}{\partial q_i(t)} w_{f,i}(t). \tag{2.25}$$

Substituting (2.23) into (2.25) yields

$$w_{f,K_j}(t) = \sum_{i \in K_j} \prod_{\substack{\ell \in K_j \\ \ell \neq i}} q_\ell(t)\, w_{f,i}(t). \tag{2.26}$$

In the case of exponential failure and repair rate

$$w_{f,K_j}(t) = \sum_{i \in K_j} \prod_{\substack{\ell \in K_j \\ \ell \neq i}} q_\ell(t)\, (1 - p_i(t))\, \lambda_i \tag{2.27}$$

where $p_i(t) = 1 - q_i(t)$ and λ_i is the failure rate for basic event i. Expression (2.27) was first proposed by Vesely [81] to be the failure density of a cut set when failure and repair rates are constant. Brown [9] later proved this rigorously.

The expected number of system failures can be computed by expression (2.24) if $\frac{\partial g(\underline{q}(t))}{\partial Q_{K_j}(t)}$ is known. We can apply the principle of inclusion-exclusion [6] and differentiate with respect to $Q_{K_j}(t)$. A less tedious calculation (and just as accurate for reliable systems) is to represent $g(\underline{q}(t))$ by the min cut upper bound

$$g(\underline{q}(t)) \leq 1 - \prod_{\ell=1}^{N_k} (1 - Q_{K_\ell}(t)) \tag{2.28}$$

and differentiate $g(\underline{q}(t))$ with respect to $Q_{K_j}(t)$,

$$\frac{\partial g(\underline{q}(t))}{\partial Q_{K_j}(t)} \leq 1 - \sum_{\substack{\ell=1 \\ \ell \neq j}}^{N_k} Q_{K_\ell}(t) + \sum_{\substack{\ell=1 \\ \ell \neq j}}^{N_k-1} \sum_{\substack{m=\ell+1 \\ m \neq j}}^{N_k} Q_{K_\ell}(t)\, Q_{K_m}(t) - \ldots \tag{2.29}$$

For reliable systems, it is common to assume

$$\frac{\partial g(\underline{q}(t))}{\partial Q_{K_j}(t)} \approx 1$$

and expression (2.24) simply becomes

$$w_{f,s}(t) = \sum_{j=1}^{N_k} w_{f,K_j}(t) \tag{2.30}$$

where $w_{f,K_j}(t)$ is given by expression (2.26) (or (2.27)) for constant failure and repair rates).

2.6.2 Distribution of Time to First Failure for a Maintained System, $F_s(t)$ - In the unrepairable case, if $\underline{F}(t) = (F_1(t), \ldots, F_n(t))$ is known, it is not difficult to compute the probability that the top event does not occur by time t (assuming statistical independence and the min cut upper bound to be accurate). Likewise, in the repairable case, it is not difficult to compute the system unavailability, a quantity depending on one point in time. In the repairable case, components can fail and be repaired many times over an interval of time and still not cause system failure. It is because of this reason that it is much more difficult to compute in the repairable case the probability that system failure does not occur over an interval of time. When the interval includes the origin t = 0, i.e., [0, t], we are interested in the distribution of time to first failure, $F_s(t)$. $F_s(t)$ may be formally defined in terms of the system reliability, $\overline{F}_s(t)$, (the probability of the nonoccurrence of the top event in [0, t]).

$$\overline{F}_s(t) = 1 - F_s(t) = P[\Psi(\underline{Y}(s)) = 0,\ 0 \leq s \leq t \mid Y_i(0) = 0 \text{ for all } i]$$

under the assumption that Ψ is coherent and that the indicator variables that are describing the occurrence of basic events in time are independent. In the following sections, when we present bounds for $F_s(t)$, we assume that the system is in perfect working order at t = 0, i.e., $q_i(0) = 0$ for each basic event i.

2.6.2.1 Approximation of $F_s(t)$, Expected Number of System Failures - We can compute a bound for $F_s(t)$ by computing the expected number of system failures as shown in (2.22). Fussell [25] took Vesely's

results and computed the failure density for the cut sets (Expression 2.27) assuming components to be at their steady state behavior at t = 0 and failure and repair distributions to be exponential.* Furthermore, if $p_i \approx 1$ we have

$$w_{f,K_j} = \sum_{i \in K_j} \prod_{\substack{\ell \in K_j \\ \ell \neq i}} q_\ell \lambda_i \tag{2.31}$$

and in conjunction with expression (2.30)

$$E[N_s(t)] = \sum_{j=1}^{N_k} \sum_{i \in K_j} \prod_{\substack{\ell \in K_j \\ \ell \neq i}} q_\ell \lambda_i t \tag{2.32}$$

since (2.31) is constant in time.

Acero [1], performed a fault tree analysis of a Boiling Water Reactor control rod drive system. Using expression (2.32) he calculated the probability of failing to insert a control rod into the reactor core in less than 11 seconds (upon demand).

2.6.2.2 Defining System Failure Rate to Find $F_s(t)$ - Vesely [81] formulated an expression for the system failure rate, $\Lambda_s(t)$, in terms of $w_{f,s}(t)$ as given in (2.24). He defines the system failure rate as

$$\Lambda_s(t) = \frac{w_{f,s}(t)}{1-g(q(t))} \tag{2.33}$$

i.e., given no failure at time t, the probability that the system fails

*Ross [64] showed that it is a conservative approximation in computing $F_s(t)$ to assume that all components are at steady state at t=0 (i.e., $q_i(0) = \frac{\tau_i}{\tau_i+\mu_i}$ for all i) when all components are working at t=0 (i.e., $p_i(0) = 1-q_i(0) = 1$ for all i).

in [t, t+dt] is $\Lambda_s(t)dt$. $\Lambda_s(t)$ is not strictly a failure rate; the above expression should be conditioned on the event, no failure in [0, t]. Vesely then defines $F_s(t)$ by

$$F_s(t) = 1 - e^{-\int_0^t \Lambda_s(t)dt}. \tag{2.34}$$

Vesely wrote the computer codes KITT-1 and KITT-2 [82] that numerically integrate $\Lambda_s(t)$ over time to estimate $F_s(t)$. Murchland claims [51] that (2.34) is no more accurate in estimating $F_s(t)$ than is the expected number of system failures. It must be noted, however, that (2.34) approaches one for large time whereas the expected number of system failures approaches infinity linearly for large time.

2.6.2.3 Finding $F_s(t)$ when Failure and Repair Distributions are Exponential - Kielson [46] has studied the Markov chain model extensively to determine $F_s(t)$. The major disadvantage of considering a Markov process is that the solution is intractable for large systems -- for a system of n components, matrices of size $2^n - 1$ by $2^n - 1$ must be inverted to find the eigenvalues and eigenvectors of the transition matrix.

Esary and Proschan [17] using the concept of association of random variables derived a bound for $F_s(t)$ in terms of distribution of time to first failure for the minimal cut sets, $F_{K_j}(t)$, assuming exponential failure and repair,

$$F_s(t) \leq 1 - \prod_{j=1}^{N_k} [1 - F_{K_j}(t)]. \tag{2.35}$$

The problem remains in determining $F_{K_j}(t)$. *Brown* [9] *derived an* expression for the Laplace transform of $F_{K_j}(t)$, denoted by $\Psi_{K_j}(s)$,

$$\psi_{K_j}(s) = \frac{1 + \sum_{r=1}^{n} (-1)^r \sum_{i_1<i_2<\ldots<i_r} \frac{s}{s + \sum_{1}^{r} (\lambda_{i_j} + \nu_{i_j})}}{1 + \sum_{r=1}^{n} \sum_{i_1<i_2<\ldots<i_r} \left(\prod_{1}^{r} \frac{\nu_{i_j}}{\lambda_{i_j}}\right) \frac{s}{s + \sum_{1}^{r} (\lambda_{i_j} + \nu_{i_j})}}$$

where $\sum_{i_1<i_2\ldots<i_r}$ denotes summation over $\binom{n}{r}$ subsets of size r from 1, ..., n, n = number of basic events in cut set K_j and λ_i and ν_i are the exponential failure and repair parameters. In general, it is very difficult to take the inverse Laplace transform to find $F_{K_j}(t)$. In comparison with the steady state process, Brown derived an upper bound for $F_{K_j}(t)$

$$F_{K_j}(t) = 1 - e^{-\left(\frac{\sum_{i=1}^{n} \mu_i}{\sigma - 1}\right)t}$$

where $\sigma = \prod_{i=1}^{n} \frac{\lambda_i + \nu_i}{\lambda_i}$.

Brown also derived a sharper bound that is more complicated and is not given here [9]. Barlow and Proschan also derived an exponential upper bound for $F_{K_j}(t)$ [7].

2.6.2.4 Other Bounds for $F_S(t)$ - In this section we limit ourselves to structures of min cut sets of order two or higher and assume that all basic events can be described in terms of an alternating renewal process. This is a simple manner of including single order cut sets in the distribution of time to first failure as shown below

$$F_s(t) \prod_{i=1}^{n} F_i(t)$$

where n equals the number of single order cut sets and $F_i(t)$ is the cumulative failure distribution of basic event i (repair has no effect in this case since failure of a single order cut set represents an absorbing state in the context of a Markov process).

2.6.2.4.1 Barlow Proschan Bound - Barlow and Proschan [7] show that if components have constant failure and decreasing repair rate then

$$F_s(t) \leq \frac{t}{1-g(\underline{A})} \sum_{i=1}^{n} (\mu_i + \tau_i)^{-1} [g(1_1, \underline{A}) - g(0_i, \underline{A})] \quad (2.36)$$

where $g(\underline{A})$ is the limiting system unavailability. The above bound is linear with respect to time. It shall be denoted as the B-P bound.

2.6.2.4.2 Steady State Upper Bound, SS, New Method to Approximate $F_s(t)$ - A new expression for $F_s(t)$ is given in this section that appears to be an upper bound for the case of constant repair rate and failure rate. The bound, called $F_{ss}(t)$, is easy to compute at the B-P upper bound. $F_{ss}(t)$ approaches one in the limit. The bound is derived in terms of assumptions that are explicitly shown without proof.

The proposed method calculates a bound for $F_s(t)$ assuming that the system is at steady state at t = 0, i.e., $q_i(0) = \frac{\tau_i}{\mu_i+\tau_i}$ for all i. Ross [63] showed that the expected number of system failures in [0, t] caused by event i occurring, $E[N_{s,i}(t)]$, in the steady is given by

$$E[N_{s,i}(t)] = \frac{t\,[g(1_i, \underline{A}) - g(0_i, \underline{A})]}{\mu_i + \tau_i} \geq F_{s,i}(t)$$

and is an upper bound for the probability that event i causes system failure exactly one time in [0, t], $F_{s,i}(t)$. Hence, the probability that i does not cause system failure in [0, t], $\bar{F}_{s,i}(t)$, is bounded from below by

$$\bar{F}_{s,i}(t) \geq 1 - \frac{t\,[g(1_i, \underline{A}) - g(0_i, \underline{A})]}{\mu_i + \tau_i}.$$

Now consider the interval $[0, \mu_i + \tau_i]$, a simple argument will show

$$\bar{F}_{s,i}(t) \geq 1 - \frac{t\,[g(1_i, \underline{A}) - g(0_i, \underline{A})]}{\mu_i + \tau_i}$$

$$\geq [1 - \{g(1_i, \underline{A}) - g(0_i, \underline{A})\}]^{\frac{t}{\mu_i+\tau_i}} \qquad (2.37)$$

since $g(1_i, \underline{A}) - g(0_i, \underline{A}) \leq 1$. Define $\Delta g_i = g(1_i, \underline{A}) - g(0_i, \underline{A})$ and recognize it to be the expected number of system failures caused by i in the steady state in $[0, \mu_i+\tau_i]$.

Assumption 1 - Assume that over each interval of time of length $\mu_i+\tau_i$, i.e., $[(n-1)(\mu_i+\tau_i), n(\mu_i+\tau_i)]$ for n = 0, 1, 2, ..., the probability that i causes system failure is independent in time, then the probability i causes system failure over each interval of time is less than or equal to $[1 - \Delta g_i]$. $\bar{F}_{s,i}(t)$ is then bounded by

$$\bar{F}_{s,i}(t) \geq [1-\Delta g_i]^{n}\,[1-\Delta g_i]^{\frac{t-n(\mu_i+\tau_i)}{\mu_i+\tau_i}} = [1-\Delta g_i]^{\frac{t}{\mu_i+\tau_i}} \qquad (2.38)$$

for

$$n(\mu_i+\tau_i) \leq t \leq (n+1)(\mu_i+\tau_i) \quad n = 0, 1, 2, \ldots$$

This bound is valid only for failure and repair distributions for which the above process of event i causing system failure is associated in time.*

Assumption 2 - Assume that probability of each component causing the system to fail is independent, then

$$\bar{F}_s(t) \geq \prod_{i=1}^{n} \bar{F}_{s,i}(t) = \prod_{i=1}^{n} [1 - \Delta g_i]^{\frac{t}{\mu_i+\tau_i}} \tag{2.39}$$

where by definition $\bar{F}_{ss}(t)$ is defined to be the steady state, upper bound given by

$$1 - F_{ss}(t) = \bar{F}_{ss}(t) = \prod_{i=1}^{n} [1 - \Delta g_i]^{\frac{t}{\mu_i+\tau_i}}. \tag{2.40}$$

In reality, basic event processes that cause the system to fail are not

---

*A performance process $\{Y_i(s), t \geq s \geq 0\}$ is associated in time if $Y_i(t')$; $Y_i(t'')$ are associated where $0 \leq t' \leq t'' \leq t$. Esary and Proschan [17] in their proof of (2.35) represented the failure and repair process of a single component $\{Y_i(s), t \geq s \geq 0\}$ in terms of a two-state Markov process. They showed the process $\{Y_i(s), t \geq s \geq 0\}$ to be associated in time. Furthermore, since the cut set indicator function, $\Psi_{K_j}(t)$, is an increasing function of its indicators, $Y_i(t)$, $\Psi_{K_j}(t)$ is associated in time since increasing functions of associated random variables are associated. Cut set indicators are associated if basic events are replicated (or independent if there is no replication). In any case, independence is a lower bound, i.e., $\bar{F}_s(t) \geq \prod_{j=1}^{m} \bar{F}_{K_j}(t)$ which implies (2.35) in failure space. If $X_1, X_2, \ldots X_n$ are associated binary random variables, $P[\prod_{i=1}^{n} X_i = 1] \geq \prod_{i=1}^{n} P[X_i = 1]$.

independent. For cut sets of order two or higher, a basic event can only cause the system to fail only if other basic events have occurred previously. Again, if one could specify the repair and failure distributions that would make basic event processes associated, then the bound in (2.39) would hold.

We can use expression (2.40) to find an upper bound for $\overline{F}_s(t)$ for a parallel system or equivalently for a cut set, K_ℓ. In this case, $F_{K_\ell}(t)$ is given by

$$\prod_{i \epsilon K_\ell} \left[1 - \prod_{\substack{j \epsilon K_\ell \\ j \neq i}} \frac{\tau_j}{\mu_j + \tau_j} \right]^{\frac{t}{\mu_i + \tau_i}} = \overline{F}_{K_\ell}(t). \tag{2.41}$$

The Esary-Proschan bound in expression (2.35) can be used to compute an upper bound for $F_s(t)$. The advantage of using expression (2.40) as opposed to expressions (2.35) and (2.41) is that one can determine d directly from (2.40) the failure density $\omega_{f,j}(t)$ given by expression (2.21). As shown in Section 2.6.2.4.5, this is useful in obtaining a more accurate bound for small time.

2.6.2.4.3 Examples of Plots of the BP and SS Upper Bounds - The Markov model is the exact solution for the distribution of time to first failure for constant failure and repair rates. Currently there is no method for finding the distribution of time to first failure for arbitrary failure and repair distributions. As an illustration, we choose two systems, assume that components have constant failure and repair rates and plot the Markov solution and the steady state upper bound as a function of time. In one case, we vary the failure and

repair rates. In the other case we assume that the system is at steady state at t = 0, i.e., $q_i(0) = \frac{\tau_i}{\tau_i+\mu_i}$. For the examples shown, it appears that the steady state upper bound is indeed an upper bound for $F_s(t)$ in the case of constant failure and repair rates.

Barlow and Proschan [5] derived the Markov distribution of time to first failure for a parallel system of two identical components with exponential failure and repair rates. The process is a birth and death stationary Markov process. The density of time to system failure is given by

$$\frac{2\lambda^2 e^{-S_1 t}}{S_2 - S_1} - \frac{2\lambda^2 e^{-S_1 t}}{S_2 - S_1}$$

where

$$S_1 = \frac{(3\lambda + \nu) + \sqrt{\lambda^2 + 6\lambda\mu + \mu^2)}}{2}$$

$$S_2 = \frac{(3\lambda + \nu) - \sqrt{\lambda^2 + 6\lambda\nu + \mu^2)}}{2}$$

where $\nu = \frac{1}{\tau}$ and $\lambda = \frac{1}{\mu}$.

In this case, the distribution of time to first failure is given by

$$\frac{2\lambda^2}{S_1(S_1 - S_2)}[1 - e^{-S_1 t}] - \frac{2\lambda^2}{S_2(S_1 - S_2)}[1 - e^{-S_2 t}].$$

This expression is plotted in figures 2.4a through 2.4f: 1) $\tau = \mu$; 2) $\tau = .1\mu$ and 3) $\tau = .01\mu$.

A table is given below for the Barlow-Proschan, B-P Bound, the Steady State bound, SS, and the Markov expression for the three cases considered above.

TABLE 2-1

Case	Distribution → B-P Upper Bound	SS Upper Bound	Markov
1) $\tau = \mu$	$F_{BP}(t) = .667t$	$F_{SS}(t) = 1.5t$	$1.207[1.-e^{-.586t}] - .207[1.-e^{-3.414t}]$
2) $\tau = .1\mu$	$F_{BP}(t) = .165t$	$F_{SS}(t) = 1.91^{1.81t}$	$1.012[1.-e^{-.156t}] - .012[1.-e^{-12.84t}]$
3) $\tau = .01\mu$	$F_{BP}(t) = .0196t$	$F_{SS}(t) = .99^{1.98t}$	$1.0002[1-e^{-.0194t}] - .0002[1-e^{-102.98t}]$

where t is expressed in units of μ.

Figures 2.4a through 2.4f show that for small time the B-P and SS upper bounds are essentially identical. For large time the SS bound remains bounded and becomes a better approximation as the expected downtime τ decreases, in particular note Fig. 2.4f. Because the B-P is linear with respect to time, it diverges for large time.

FIG. 2.4 Comparisons of Upper Bounds of the Distribution of Time to First Failure for Two Identical Components in Parallel for Various Values of μ, τ and t.

μ = Mean Time to Failure

τ = Mean Time to Repair

t = Time, Expressed in Units of μ

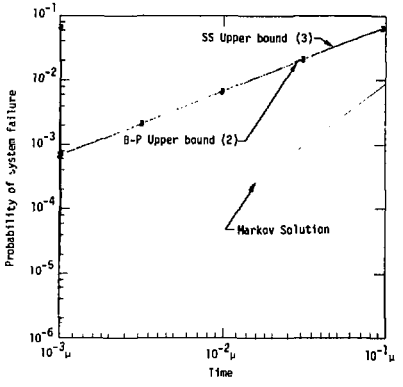


FIG. 2.4a
$\tau = \mu$
Case of Small Time

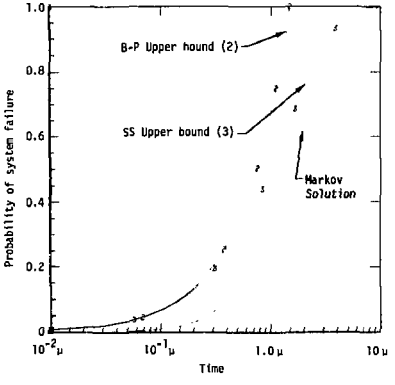


FIG. 2.4b
$\tau = \mu$
Case of Large Time

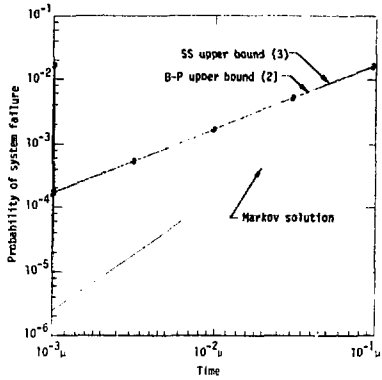


FIG. 2.4c
$\tau = .1\mu$
Case of Small Time

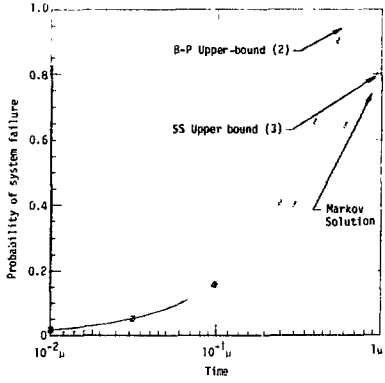


FIG. 2.4d
$\tau = .1\mu$
Case of Large Time

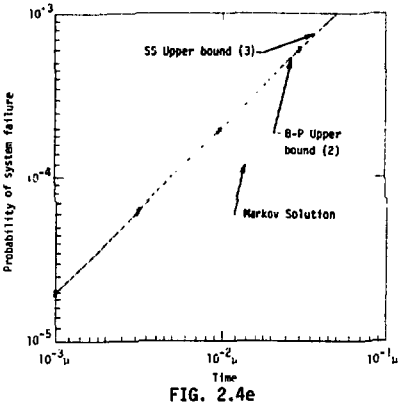


FIG. 2.4e

$\tau = .01\mu$

Case of Small Time

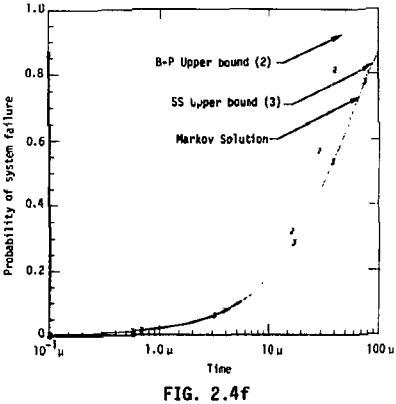


FIG. 2.4f

$\tau = .01\mu$

Case of Large Time

2.6.2.4.4 A Better Approximation for Small Time -

We see in Figures 2.4a, c, and e that both the B-P Bound and the SS Bound considerably overpredict system failure. For small time, the expected number of failures is a good approximation for system failure.

In Figures 2.5a and b, we plot the expected number of system failure as a function of time for the case $\tau = .1\mu$ assuming at $t = 0$, $p_1(0) = p_2(0) = 1$.

We see in Figure 2.5a that for small t, the expected number of system failures, $E[N_s(t)]$, is an excellent approximation. However, as shown in Figure 2.5b, it is asymptotically linear and a poor approximation for large time.

FIG. 2.5 Comparison of the Steady State Upper Bound and the Expected Number of System Failures, $E[N_s(t)]$, with the Markov Solution

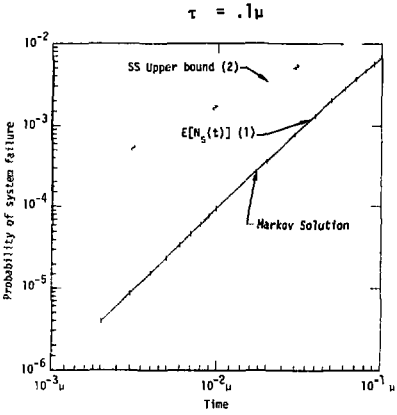


FIG. 2.5a
Case of Small Time

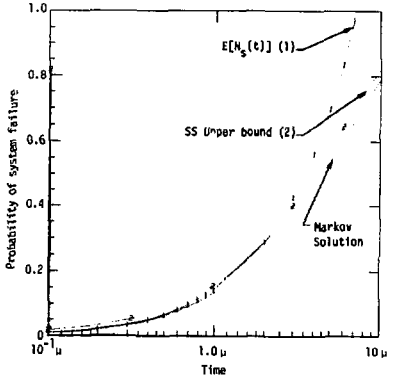


FIG. 2.5b
Case of Large Time

2.6.2.4.5 The T*(Tee-Star) Method - In some cases, it might be desirable to have a "good" approximation for small and large times, i.e., it might be desirable to approximate $F_s(t)$ at large time. This can be done by determining the time at which the steady-state rate of system breakdown is a better approximation than the rate predicted by the expected number of failures. This time will, in general, be different for each component in the system if the failure and repair distributions are different.

Define $\bar{F}_{s,i}(t)$ as given in expression (2.37) by

$$p_i(t) = \bar{F}_{s,i}(t) = [1 - \Delta g_i]^{\frac{t}{\mu_i + \tau_i}}$$

then $\bar{F}_{ss}(t)$ in expression (2.40) is given by

$$F_{ss}(t) = \prod_{i=1}^{n} p_i(t)$$

by the chain rule of differentiation

$$\frac{d\bar{F}_{ss}(t)}{dt} = \frac{\partial \bar{F}_{ss}(t)}{\partial p_1(t)} \frac{dp_1(t)}{dt} + \ldots + \frac{\partial \bar{F}_{ss}(t)}{\partial p_n(t)} \frac{dp_n(t)}{dt} \qquad (2.42)$$

noting that

$$\frac{dF_{ss}(t)}{dt} = - \frac{d\bar{F}_{ss}(t)}{dt} .$$

We can identify the rate that event i causes the first system failure from expression (2.42) as

$$\omega^{ss}_{f,i}(t) = \frac{-\partial \bar{F}_{ss}}{\partial p_i(t)} \frac{dp_i(t)}{dt} \qquad (2.43)$$

which is analogous to expression (2.21). Performing the differentiation, (2.43) becomes

$$\omega_{f,i}^{ss}(t) = \frac{-\ln[1-\Delta g_i]}{\mu_i+\tau_i} F_{ss}(t). \qquad (2.44)$$

If we plot expression (2.21) and expression (2.43) versus time in the manner shown below

$$\omega_{f,i}(t) \text{ vs. } t$$

and

$$\{1 - E[N_{s,i}(t)]\}\ \omega_{f,i}^{ss}(0) \text{ vs. } t$$

where

$$E[N_{s,i}(t)] = \int_0^t \omega_{f,i}(t)dt$$

we can find the time, designated as T_i^*, when the steady rate of breakdown caused by component i becomes a better approximation than the failure density given in (2.21) for computing $F_s(t)$.

This value for a parallel system of two components with $\tau = .1\mu$ is approximately $.2\mu$ as shown in Figure 2.6. The distribution of time to first failure according to the T* method is given by

$$F_s^{T^*}(t) = \sum_{i=1}^{n} g_{s,i}(t) \text{ where n is the number of components} \qquad (2.45)$$

where

$$g_{s,i}(t) = \begin{cases} E[N_{s,i}(t)] & t \le T_i^* \\ E[N_{s,i}(T_i^*)] + (1-E[N_{s,i}(T_i^*)]) \cdot \int_{T_i^*}^{t} d\omega_{f,i}^{ss}(t-T_i^*) & t > T_i^*. \end{cases}$$

An example of the T* method is given in Figure 2.7 for $\tau = .1\mu$. The greatest deviation between $F_s^{T^*}(t)$ and the Markov solution is 5% for all t.

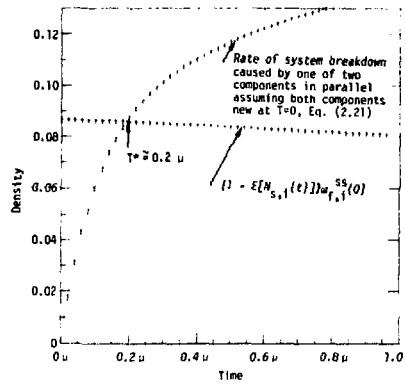


FIG. 2.6 Density Plots Determining T*

$\tau = .1\mu$

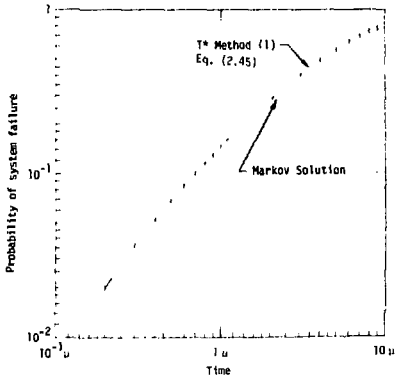


FIG. 2.7 Plot of Results of T* Method -- An Upper Bound on the Distribution of Time to First Failure

The T* Method can be useful if the failure density, $w_f(t)$, is known, such as in the cases of exponential failure and repair, exponential failure and gamma repair, and gamma failure and repair distributions. Consult reference [3] for these distributions. The T* Method is well suited for computer applications.

2.6.2.4.6 A More Complex Example Illustrating Behavior of Proposed Method - In this section, we find the distribution of time to first failure by the Markov method for a more complex system. In one case, we find $F_s(t)$ by the Markov model, assume the system to be at steady state at t = 0, and compare the plot of this distribution with the steady state upper bound. In the other case, we assume all components to be new at t = 0, and compare the Markov solution with the T* method.

The system considered is a two-out-of-three system in parallel with a single component as shown in Fig. 2.8.

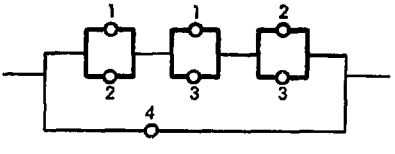


FIG. 2.8 System 2-C

We assume that all components are maintained with $\mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu$, and $\tau = \tau_1 = \tau_2 = \tau_3 = \tau_4 = .1\mu$ where as before, μ represents mean time to failure and τ represents mean time to repair. Let Y_i be the indicator variable

$$Y_i = \begin{cases} 1 \text{ if component } i \text{ is failed} \\ 0 \text{ otherwise} \end{cases}$$

and let the ordered pair (x_1, x_2) represent $(Y_4, \sum_{i=1}^{3} Y_i)$, a possible system state. There are seven states for system 2-C as shown in Figure 2.9, similar to Figure 1.4 of Chapter One.

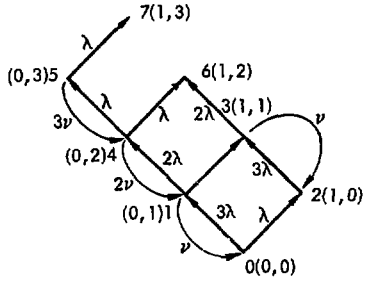


FIG. 2.9 Transition State Diagram

The transition matrix is shown below where $\lambda = \frac{1}{\mu}$ and $\nu = \frac{1}{\tau}$.

We recognize that states 6 and 7 are absorbing states, i.e., the transition rates from states 6 and 7 are zero as indicated in the diagonal of the transition matrix. The distribution of time to first failure is given by

$$F_s(t) = P_6(t) + P_7(t)$$

where, as in Chapter One, $P_i(t)$ represents the probability that the system is in state i at time t. We will consider two possible solutions: first

we assume that all components are working at t = 0, i.e., $P_1(0) = 1$. $F_s(t)$ has six negative, real eigenvalues and is given by $F_{s,0}(t)$ as

$$F_{s,0}(t) = 1 - 1.007043e^{-.057208t} - .001724e^{-26.00695t} + .000410e^{-22.101247t} - .000010e^{-11.003548t} - .000090e^{-34.025200t} + .008799e^{-11.805852t} \quad (2.46)$$

where t is expressed in units of μ.

We now consider system operation at steady state. The first column in Fig. 2.10 gives us the probability that the system is in state i at t = ∞, i.e., $P_i(\infty)$.

$P_i(\infty)$		0	1	2	3	4	5	6	7
p_∞^4	0	-4λ	ν	ν	0	0	0	0	0
$3p_\infty^3 q_\infty$	1	3λ	$-(3\lambda+\nu)$	0	ν	2ν	0	0	0
$p_\infty^3 q_\infty$	2	λ	0	$-(3\lambda+\nu)$	ν	0	0	0	0
$3p_\infty^2 q_\infty^2$	3	0	λ	3λ	$-(2\lambda+2\nu)$	0	0	0	0
$3p_\infty^2 q_\infty^2$	4	0	2λ	0	0	$-(2\lambda+2\nu)$	3ν	0	0
$p_\infty q_\infty^3$	5	0	0	0	0	λ	$-(\lambda+3\nu)$	0	0
$3p_\infty q_\infty^3$	6	0	0	0	2λ	λ	0	0	0
q_∞^4	7	0	0	0	0	0	λ	0	0

FIG. 2.10 Asymptotic State Probabilities and Transition Matrix

P_∞ and q_∞ represent the asymptotic availability and unavailability at $t = \infty$ of all components in the system. Now let us assume steady state operation at $t = 0$. If we assume that states 6 and 7 are not occupied at $t = 0$, states 1 through 5 are occupied with probability

$$P_i(0) = \frac{P_i(\infty)}{1-3p_\infty q_\infty^3-q_\infty^4} \quad \text{for } i = 1, 2, 3, 4 \text{ and } 5 \qquad (2.47)$$

and

$$P_6(0) = P_7(0) = 0$$

where

$$P_\infty = \frac{1}{1.1} = .909 \text{ and } q_\infty = \frac{.1}{1.1} = .091.$$

In other words, if we are given that the system is up in the steady state, the probability of the system occupying a state is given by (2.47). The asymptotic solution for $F_s(t)$ in this asymptotic case is given by $F_{s,\infty}(t)$ as

$$\begin{aligned} F_{s,\infty}(t) = 1 &- .999970e^{-.057208t} - .000102e^{-26.00695t} \\ &+ 0.(10^{-6})e^{-22.101247t} + 0.(10^{-6})e^{-11.003548t} \\ &- .000002e^{-34.025200t} - .000194e^{-11.805852t} \end{aligned} \qquad (2.48)$$

where again t is expressed in units of μ. The steady state upper bound, $F_{ss}(t)$, is given by

$$\bar{F}_{ss}(t) = \prod_{i=1}^{4} [1-\Delta g_i]^{\frac{t}{\mu_i+\tau_i}} = \left[(.985)^{\frac{t}{1.1}}\right]^3 (.977)^{\frac{t}{1.1}}$$

and

$$F_{ss}(t) = 1 - e^{-0.062716t}. \qquad (2.49)$$

Note the simplicity of expression (2.49) as compared with (2.46) or (2.48).

We see in Figure 2.11, that the steady state upper bound, (2.49) and the asymptotic Markov solution, (2.48), exhibit nearly the same behavior At large time $F_{SS}(t)$ is slightly greater than $F_{S,\infty}(t)$ since for large time, expression (2.48) shows that $F_{S,\infty}(t) \approx 1 - e^{-.057208t}$ which is always less than $F_{SS}(t)$ as given in (2.49). We might conjecture, at this point, that the assumption of independence, assumptions 1 and 2 in Section 2.6.2.4.2 leads to the slight overprediction.* We see in Figure 2.11 that the steady state upper bound considerably overpredicts

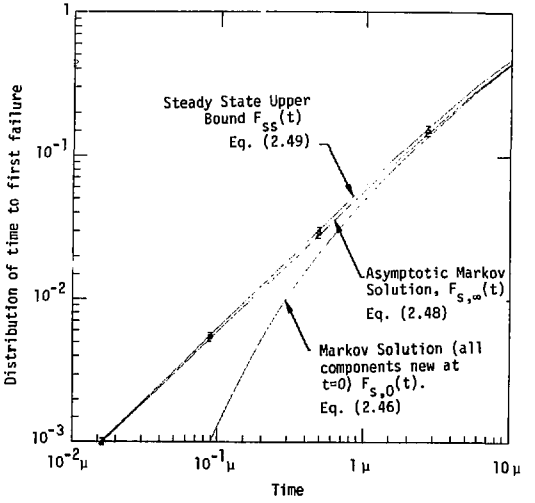


FIG. 2.11 Comparison of Steady State Upper Bound with Markov Solutions

system failure for small time if all components are new at $t = 0$.

*This same assumption leads to the slight overprediction of the min cut upper bound as shown in Figure 2.1.

A better approximation for small time can be obtained from the T* Method. Density plots similar to Figure 2.6 show that $T_i^* = .3\mu$ for all components. The T* approximation and $F_{s,0}(t)$ are plotted versus time in Figure 2.12.

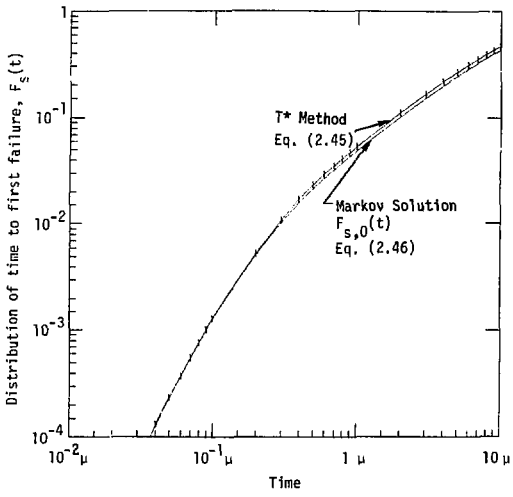


FIG. 2.12 Comparison of T Method with Markov Solution*

Figure 2.12 exhibits the same behavior as Fig. 2.7. The T* Method is as accurate as the expected number of failures for small time, i.e., $t < .3\mu$, and slightly overpredicts system failure for large time, i.e., $t > .3\mu$.

2.6.3 Mean Time to First Failure for a Maintained System - Use of the steady state upper bound provides a simple and direct way of computing the mean time to first failure, MTFF, for a maintained system. The MTFF is given by

$$\text{MTFF} = \int_0^\infty \bar{F}_s(t)dt.$$

Integration of this expression yields a lower bound for the MTFF and is simply given by

$$\text{MTFF} > \int_0^\infty \bar{F}_{ss}(t)dt = \frac{1}{\sum_{i=1}^{n} \frac{\ln(1-\Delta g_i)}{\mu_i+\tau_i}} \qquad (2.50)$$

when $F_s(t)$ is approximated by the steady state upper bound. Recall that $\Delta g_i = [g(1_i, \underline{q}) - g(0_i, \underline{\bar{A}})]$. Furthermore, if there are m components in single order cut sets with exponential life distributions, than expression (2.50) becomes

$$\text{MTFF} > \frac{1}{\sum_{i=1}^{n} \frac{\ln(1-\Delta g_i)}{\mu_i+\tau_i} + \sum_{j=1}^{m} \lambda_j}$$

where $g = E[\Psi(\underline{Y}(t))]$ and $\Psi(\underline{Y}(t))$ is the structure function for the union of all min cut sets of order two and higher. The mean time to first failure is computed for the two systems considered previously and is given in Table 2-2.

We see that the fractional downtime decreases, the SS upper bound becomes a better approximation (as expected from the behavior shown in Figures 2.4b, 2.4d and 2.4f).

TABLE 2-2

Mean Time to First System Failure

System	$\tau(\mu)$	MTFF$[\bar{F}_s(t) = \bar{F}_{ss}(t)]$	MTFF$[\bar{F}_s(t) = \bar{F}_{Markov}(t)]$
Fig. 2.4a	$\tau = \mu$	1.44μ	2.00μ
Fig. 2.4c	$\tau = .1\mu$	5.79μ	6.49μ
Fig. 2.4e	$\tau = .01\mu$	50.30μ	51.56μ
Fig. 2.8	$\tau = .1\mu$	15.94μ	17.61μ

A better approximation to the MTFF can be calculated from the T* Method. The MTFF in this case is given by

$$\text{MTFF} = \int_0^\infty \sum_{i=1}^{n} (1-g_{s,i}(t))dt$$

where $g_{s,i}(t)$ is given by expression (2.45)

2.7 Other Reliability Questions Pertinent to Fault Tree Analysis

We may often wish to incorporate redundancy in order to increase the reliability or safety of the system. Often the reliability of the connecting elements (or quasi static components) is, however, not considered. As shown in the following subsection, this can lead to erroneous conclusions regarding the most reliable or safe system design.

We also consider in this section the probabilistic evaluations of priority AND gates in which the order of occurrence of the input events is relevant in causing the output event to occur. Finally, in the last subsection, an expression for the limiting unavailability of a component due to secondary failure mechanisms is derived.

2.7.1 *Connector Reliability When Considering Redundancy**- There are basically two ways of upgrading a system design to improve its reliability; we can incorporate redundancy either at the system or at the component level. For example, in the series system shown in Fig. 2.13, system redundancy is accomplished by simply placing an identical system in parallel, as shown in Figure 2.14, where the primes denote components

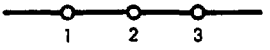


FIG. 2.13 System 2-D

identical to the unprimed components. (Let us for a moment neglect valves that are shown in Fig. 2.14). For component redundancy, we simply

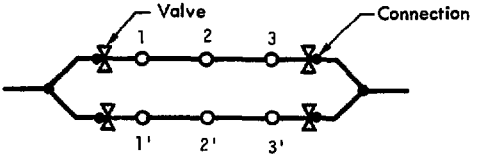


FIG. 2.14 System Redundancy for System 2-D

place an identical component in parallel with every component in the system, as shown in Fig. 2.15.

*Example in this section due to D. Haasl [40].

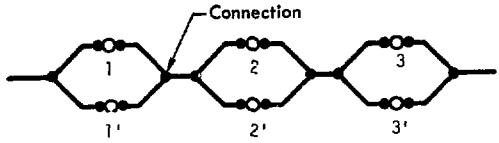


FIG. 2.15 Component Redundancy for System 2-D

Barlow and Proschan [6] show that when we consider active components only, the reliability of the system when replicated at the component level is always greater than the reliability of a system replicated at the system level (one exception is parallel systems in which reliabilities are equal). When quasi static components or connections are also considered, this result may not be true. For example, if system 2-D were a hydraulic system, then a pipe rupture anywhere in the system shown in Figure 2.15 is catastrophic. However, as shown in Fig. 2.14, valves may be placed in each redundant leg to isolate pipe ruptures that may occur in either leg. There are nine minimal cut sets of order two involving failure of active components in Fig. 2.14 and only three min cut sets for Fig. 2.15. However, in Fig. 2.15 there are 16 pipe connections whose rupture is catastrophic and only 6 in Fig. 2.14. The failure rate of an active component is of the order 10^{-5}/hr. The failure rate is approximately three orders of magnitude less for quasi static components, i.e., $\sim 10^{-8}$/hr.*

*William Vesely [78] reports that actual failure rates of quasi static components may be one to two orders of magnitude higher than those reported in the literature. Quasi static components commonly fail on demand. The time over which the failure actually occurs may be significantly smaller than the reported time on which the failure rate is based.

Table 2-3 lists the probabilities of failure associated with each system failure mode.

TABLE 2-3

Failure Contribution Probabilities

System	Active Component Failure Contribution	Pipe Rupture Contribution	Total
Fig. 2.14	9×10^{-10}/hr	6×10^{-8}/hr	$\sim 6.1 \times 10^{-8}$/hr
Fig. 2.15	3×10^{-10}/hr	1.6×10^{-7}/hr	$\sim 1.6 \times 10^{-7}$/hr

We see that the failure of quasi-static components dominates in the calculation of the probability of system failure.*

For electrical systems, component redundancy generally results in more reliable arrangements than system redundancy because an open circuit at a connection in electrical circuits is not as catastrophic (in general) as a pipe rupture in hydraulic systems. However, in may cases, physical isolation at the system level is also preferred for electrical systems in order to allow for functional diversity and minimize the likelihood of common mode failures associated with proximity of equipment.

2.7.2 Priority AND Gates - A priority AND gate is logically equivalent to an AND gate with the added stipulation that the input events must occur in a specific order. If all input events have an infinite fault duration time and all input probabilities $F_1(t)$ are equal for all time, then the probability of the output event, as a function of time,

*A more in-depth analysis would also have to consider rupture of the valves.

$Q_n(t)$, is given by

$$Q_n(t) = \frac{[F(t)]^n}{n!}$$

where n is the number of input events and $\frac{1}{n!}$ is the combinational factor specifying the probability associated with one outcome sequence. For the general case where repair is not allowed, $Q_n(t)$ is given by

$$Q_n(t) = \int_{t_n}^{t} dF_n(t) \ldots \int_{t_1}^{t_2} dF_2(t) \int_{0}^{t_1} dF_1(t).$$

Aber [28] gave the following result for $Q_n(t)$ when all basic events have an exponential life distribution

$$Q_n(t) = \left[\prod_{j=1}^{n} \lambda_j\right] \left[\sum_{K=0}^{n} \frac{e^{a_K t}}{\prod_{\substack{J=0 \\ J\neq K}}^{n} (a_K - a_J)}\right]$$

where

$$a_0 = 0$$

$$a_J = \sum_{j=1}^{J} \lambda_j \text{ for } J>0$$

$$a_K = \sum_{k=1}^{K} \lambda_k \text{ for } K>0.$$

Because the output event of a priority AND gate is caused by a particular sequence occurring in time, priority AND gates do not obey the laws of conditional probability, i.e., the relative frequency interpretation does not hold. For example, for a priority AND gate with two input events, A and B,

$$P(A/B)P(B) \neq P(B/A)P(B).$$

2.7.3 Calculation of System Unavailability for Fault Trees with Secondary Failures - For maintained systems, the system unavailability cannot be calculated by conventional means for fault trees that contain secondary failures.* Secondary failures are not statistically independent failures. Failure, in this case, is caused by environmental or operational stress placed on the component. For example, in the fault tree in Fig. 1.17b, it is the switch 1 or 2 contacts failing in the closed position that causes an overrun of the battery. Repair (i.e., recharging of the battery) takes place due to failure mechanisms that are external rather than internal to the battery. Whether the component fails due to secondary or primary causes, the end result is the same. The component is in a failed state and must be repaired (or replaced) to return the system to a normal operating state. In this section, we derive an expression for the limiting unavailability of a component due to secondary failure causes.

As shown in Figure 1.13, inhibit gates are used to describe secondary failures in fault trees. We make the assumption that the probability of the inhibit condition (i.e., the conditional event) is constant in time. This probability shall be denoted as I_i for component i. We treat each secondary event as a module in the fault tree.

Notation:

1. $x_i^M(t)$ is defined as the structure function for the module M_i that describes the secondary failure of component i.
2. $E[x_i^M(t)] \overset{\text{def}}{=} g^{Mi}(\underline{A}(t))$.

*The author became aware of this fact in conversations with Jerry Fussell[26].

3. $G_i(t)$ is the repair distribution of component i.

4. $\Delta g_j^{M_i}(\underline{\bar{A}}(t)) = g^{M_i}(1_j, \underline{\bar{A}}(t)) - g^{M_i}(0_j, \underline{\bar{A}}(t))$.

5. $\Delta g_j^{M_i}(\underline{\bar{A}}) = g^{M_i}(1_j, \underline{\bar{A}}) - g^{M_i}(0_j, \underline{\bar{A}})$

where $\bar{A}$ is the limiting unavailability.

We are also assuming that each basic event in M_i can be described in terms of an alternating renewal process.

Derivation:

For component i to be down at time t due to secondary causes, a component (or basic event) must have caused i to fail prior to t (say at t') and repair must not have taken place in [t', t]. Any basic event contained in the module M_i can cause i to fail.

The probability that the component i is down due to a secondary failure at time t is given by $\bar{A}_i^S(t)$ as

$$\bar{A}_i^S(t) = I_i \int_0^t \sum_{j \in M_i} \Delta g_j^{M_i}(\underline{\bar{A}}(t)) \, [1-G_i(t-t')] w_{f,j}(t')dt'. \quad (2.51)$$

In (2.51), we are making the conservation assumption of neglecting the simultaneous occurrence of two or more min cut sets in $\chi_i^M(t)$ when component i is down for repair. Therefore, (2.51) is an upper bound. We now find the limiting value of (2.51) as $t \to \infty$ to obtain the limiting unavailability. To do this we use Laplace transforms.

$$\bar{A}_i^S = \lim_{t\to\infty} \bar{A}_i^S(t) = \lim_{s\to 0} s \, \tilde{\bar{A}}_i^S(s). \quad (2.52)$$

First let us find the Laplace transform of the renewal density $w_{f,j}(t)$ given as $\tilde{w}_{f,j}(s)$. If component j has failure density, $f_j(t)$ and repair density $g_i(t)$, (2.15) can be written as

$$w_{f,j}(t) = f_j(t) + \int_0^t dx\, w_{f,j}(x) \int_0^{t-x} g_j(t-x-t')\, f_j(t')dt' \quad (2.53)$$

which implies by the convolution theorem for Laplace transforms

$$\tilde{w}_{f,j}(s) = \tilde{f}_j(s) + \tilde{w}_{f,j}(s)\, \tilde{g}_j(s)\, \tilde{f}_j(s)$$

or

$$\tilde{w}_{f,j}(s) = \frac{\tilde{f}_j(s)}{1-\tilde{f}_j(s)\tilde{g}_j(s)}\,. \quad (2.54)$$

Next, we want fo find $\tilde{f}_j(s)$ and $\tilde{g}_j(s)$ for small s. By definition

$$f_j(t) = -\frac{d\bar{F}_j(t)}{dt} \text{ and } g_j(t) = -\frac{d\bar{G}_j(t)}{dt}$$

where $\bar{F}_j(t) = 1 - F_j(t)$ and $\bar{G}_j(t) = 1 - G_j(t)$ which implies

$$\tilde{f}_j(s) = -[s\tilde{\bar{F}}_j(s) - \bar{F}_j(0)] \text{ and } \tilde{g}_j(s) = -[s\tilde{\bar{G}}_j(s) - \bar{G}_j(0)]$$

where by the definition of the Laplace transform

$$\tilde{\bar{F}}_j(s) = \int_0^\infty \bar{F}_j(t)\, e^{-st}\, dt \text{ and } \tilde{\bar{G}}_j(s) = \int_0^\infty \bar{G}_j(t)\, e^{-st} dt. \quad (2.55)$$

Recall that

$$\mu_j = \int_0^\infty \bar{F}_j(t)dt, \text{ likewise } \tau_j = \int_0^\infty \bar{G}_j(t)dt. \quad (2.56)$$

For small s expressions (2.55) and (2.56) imply that

$$\mu_j \cong \tilde{\bar{F}}_j(s) \text{ and } \tau_j \cong \tilde{\bar{G}}_j(s) \quad (2.57)$$

and

$$\tilde{f}_j(s) = 1 - \mu_j s \text{ and } \tilde{g}_j(s) = 1 - \tau_j s. \tag{2.58}$$

Now take the Laplace transform of (2.51)

$$\bar{A}_i^S(s) = I_i \sum_{j=1}^{n} \Delta g_j^{M_j}(\bar{A}(s)) \frac{\tilde{f}_j(s)}{1-\tilde{f}_j(s)\tilde{g}_j(s)} \left[\frac{1}{s} - \frac{\tilde{g}_j(s)}{s}\right]. \tag{2.59}$$

Substituting (2.58) into (2.59), we get

$$\bar{A}_i^S(s) = I_i \sum_{j=1}^{n} \frac{\Delta g_j^{M_j}(\bar{A}(s))[1 - \mu_j s][\frac{\tau_j s}{s}]}{1-(1-\mu_j s)(1-\tau_j s)}. \tag{2.60}$$

Using expression (2.52) and L'Hospital's Rule, the limiting unavailability of component i due to secondary failure causes is given by

$$\bar{A}_i^S = I_i \sum_{j \in M_i} \Delta g_j^{M_j}(\bar{A}) \frac{\tau_j}{\mu_j+\tau_j}. \tag{2.61}$$

When calculating the limiting system unavailability, we simply remove all secondary failures from the fault tree and estimate the unavailability of component i as

$$\bar{A}_i = \frac{\tau_i}{\mu_i+\tau_i} + I_i \sum_{j \in M_i} \Delta g_j^{M_j}(\bar{A}) \frac{\tau_j}{\mu_j+\tau_j} \tag{2.62}$$

where it is recognized that the first term in (2.62) is simply the limiting unavailability of component i due to internal or primary causes.

2.8 Reliability Quantification Techniques Used in the Reactor Safety Study

As described in Section 1.8, the Study defined reactor accidents in terms of accident sequences, schematically represented as

Accident Sequence	=	Initiating Event	x	System Failure	x	Containment Failure Mode	
AS		A	x	B	x	C.	(2.63)

In the study, top level system fault trees were required to define the combination of failure of engineered safeguard systems (ESS) that cause a containment failure and in turn leads to a certain radiological release. The initiating event served as an initial condition for top level system fault trees. Accident sequences were quantified using the laws of conditional probability, i.e., in terms of (2.64)

$$P(AS) = P(A)\ P(B|A)\ P(C|B \cdot C) \tag{2.64}$$

since in (2.64) the outcome of each event depends upon events that have occurred previously in the sequence. In the following subsections, we discuss the methods for obtaining the probability of each term in (2.64). In particular, we concentrate on obtaining system failure probabilities, $p(B|A)$ by the fault tree technique. The study showed that testing, maintenance and human error contributed greatly to the downtime of critical ESS components. System failure probabilities computed by the Study were in some cases orders of magnitude greater than those previously calculated by the nuclear vendors.

2.8.1 Initiating Events - The first type of initiating events considered were pipe breaks in the primary coolant system. Since the ESS requirements vary with the size of the break, pipe breaks of different sizes were assumed as initiating events. Other initiating events considered were (1) catastrophic rupture of the pressure vessel, (2) unchecked system interface conditions and (3) transient events that

are expected to occur, such as a turbine trip and loss of offsite power.

The Study examined nuclear as well as industrial and other data sources and estimated the probability of these initiating events and other confidence limits. For pipe ruptures, the study compiled the following data.

TABLE 2-4

Pipe Break Data Compiled by the

Reactor Safety Study

Pipe Rupture Size (Inches in Dia.)	LOCA Initiating Rupture Rates (Per Plant Per Year)	
	90% Range	Median
1/2 - 2	$1 \times 10^{-4} - 1 \times 10^{-2}$	1×10^{-3}
2 - 6	$3 \times 10^{-5} - 3 \times 10^{-3}$	3×10^{-4}
>6	$1 \times 10^{-5} - 1 \times 10^{-3}$	1×10^{-4}

2.8.2 Fault Tree Development and Quantification - Technical specifications by NRC require that all active components in the ESS be redundant ("single failure" criterion), including all instrument channels that initiate ESS action following a LOCA. Fault trees that describe failure of active components within these systems should contain min cut sets of order two or higher. However, in the following sections, we show that single order cut sets do exist in these system fault trees. Furthermore, we show commonality between basic events in cut sets of order two and higher that violates the assumption of independence of the basic events.

We first consider the iterative process by which fault trees were generated for the Study before discussing quantification techniques.

2.8.2.1 Fault Tree Construction - The analysts had to acquire a thorough understanding of the systems being analyzed. This was partially accomplished by examining detailed sets of design drawings and specifications, safety analysis reports, flow diagrams, process and instrumentation diagrams, equipment location diagrams, control system logic diagrams, electrical schematics, and emergency, operating, and testing and maintenance procedures. In addition, the fault-tree analysts made inspection trips to the plant site to verify system design and layout and to inspect the installed system hardware.

Fault tree construction proceeded in two steps; first detailed fault trees were drawn. Consideration was given to system interface conditions, common power sources, common instrumentation and detectors. As the analyst became more familiar with his system, he incorporated the more subtle aspects of system behavior in his fault tree. The fault trees "grew" and became very complex and difficult to evaluate. In the second step, fault trees were simplified by elimination of negligible contributions. In this reduction process, the following min cut sets were thought to be most important.

1. single passive faults
2. single active faults
3. double active faults

and were retained. In some cases, third order cut sets were retained.*

*For the PWR electric power fault trees, the most significant contribution to loss of electric power was the triple cut set, "loss of offsite power and two diesel generators fail to start". In another case, the BWR scram system fault trees contained no single or second order cut sets; quantification was based on third and higher order cut sets.

2.8.2.2 System Unavailability - The engineered safeguard systems are standby safety systems and the Study was concerned with all the factors that could cause these systems to fail upon demand. In particular, their efforts were directed to two major areas, (1) the possible existence of undetected failures for extended time periods caused by either human or hardware related faults and (2) the system downtime due to scheduled maintenance or testing. Their conclusion was that four major factors contributed to system unavailability:

1. random hardware failures
2. periodic testing
3. maintenance
4. human error.

We now consider each one of these factors in order and choose the containment spray injection system and the low pressure injection systems given in Fig. 1.5 as examples to illustrate the calculations.

2.8.2.2.1 Hardware Contribution Q - In the event of a LOCA, the containment spray injection system, CSIS, and the low pressure injection system, LPIS, start on two signals, the consequence limiting signal (CLS) and the safety injection signal (SIS). When the containment pressure reaches 1 psig, the CLS initiates action that opens the motor operated valves, V1, V2, V3 and V4 and start pumps, P1 and P2. The SIS detects low coolant pressure and initiates action that starts low pressure injection pumps, P3 and P4. The CLS can also start the low pressure injection system. With these active components we are concerned with two types of failure, (1) at t=0, failure to change state and (2) failure to continue operation given a successful start. Based on the data collected for the Study, point estimates based on the

We first consider the iterative process by which fault trees were generated for the Study before discussing quantification techniques.

2.8.2.1 Fault Tree Construction - The analysts had to acquire a thorough understanding of the systems being analyzed. This was partially accomplished by examining detailed sets of design drawings and specifications, safety analysis reports, flow diagrams, process and instrumentation diagrams, equipment location diagrams, control system logic diagrams, electrical schematics, and emergency, operating, and testing and maintenance procedures. In addition, the fault-tree analysts made inspection trips to the plant site to verify system design and layout and to inspect the installed system hardware.

Fault tree construction proceeded in two steps; first detailed fault trees were drawn. Consideration was given to system interface conditions, common power sources, common instrumentation and detectors. As the analyst became more familiar with his system, he incorporated the more subtle aspects of system behavior in his fault tree. The fault trees "grew" and became very complex and difficult to evaluate. In the second step, fault trees were simplified by elimination of negligible contributions. In this reduction process, the following min cut sets were thought to be most important.

1. single passive faults
2. single active faults
3. double active faults

and were retained. In some cases, third order cut sets were retained.*

*For the PWR electric power fault trees, the most significant contribution to loss of electric power was the triple cut set, "loss of offsite power and two diesel generators fail to start". In another case, the BWR scram system fault trees contained no single or second order cut sets; quantification was based on third and higher order cut sets.

2.8.2.2 System Unavailability - The engineered safeguard systems are standby safety systems and the Study was concerned with all the factors that could cause these systems to fail upon demand. In particular, their efforts were directed to two major areas, (1) the possible existence of undetected failures for extended time periods caused by either human or hardware related faults and (2) the system downtime due to scheduled maintenance or testing. Their conclusion was that four major factors contributed to system unavailability:

1. random hardware failures
2. periodic testing
3. maintenance
4. human error.

We now consider each one of these factors in order and choose the containment spray injection system and the low pressure injection systems given in Fig. 1.5 as examples to illustrate the calculations.

2.8.2.2.1 Hardware Contribution Q - In the event of a LOCA, the containment spray injection system, CSIS, and the low pressure injection system, LPIS, start on two signals, the consequence limiting signal (CLS) and the safety injection signal (SIS). When the containment pressure reaches 1 psig, the CLS initiates action that opens the motor operated valves, V1, V2, V3 and V4 and start pumps, P1 and P2. The SIS detects low coolant pressure and initiates action that starts low pressure injection pumps, P3 and P4. The CLS can also start the low pressure injection system. With these active components we are concerned with two types of failure, (1) at t=0, failure to change state and (2) failure to continue operation given a successful start. Based on the data collected for the Study, point estimates based on the

log normal distribution were obtained

Q pump (failure to start) -- 10^{-3} per demand

Q pump (failure to run, given start) -- 3×10^{-5}/hr.

Q valve (motor operated, failure to open or close) -- 10^{-3} per demand

Q valve (inadvertently opens or closes at $t > 0$) -- 10^{-6} per hour.

2.8.2.2.2 Maintenance Contribution, M - Preventive maintenance is required to keep the failure rates constant over the 30-year plant life. The Study assumed scheduled maintenance of the CSIS and LPIS pumps to be performed on an interval ranging from 1 to 12 months, with a log normal mean of 4.5 months. The maintenance duration is assumed to be between 30 minutes and 24 hours, with a log normal mean of 7.1 hours.* The average unavailability of one leg of the CSIS or LPIS due to maintenance is then $7.1/(720 \times 4.5) = 2.2 \times 10^{-3}$. In general, the interval unavailability due to maintenance was calculated from the relation

M = f(acts per month) x t(hours per month)/720 (hours per month)

where f is the maintenance frequency and t is the length of duration of the maintenance act. A maintenance contribution is calculated only for hardware requiring isolation from the system during maintenance.

2.8.2.2.3 Testing Contribution, T - Technical specifications by NRC require that CSIS and LPIS be tested once a month. Each leg of the CSIS when tested is effectively disabled. Tests of each

*The upper limit of 24 hours is due to the fact that technical specifications require plant shutdown if maintenance lasts more than 24 hours.

CSIS pump take at least 15 minutes and technical specifications require plant shutdown if the CSIS pump is unavailable for more than four hours. Based on these two extremes, the log-normal mean test duration is 1.4 hours. The unavailability of each CSIS leg is then $1.4/720 = 1.9 \times 10^{-3}$. LPIS pumps have an override capability permitting automatic return of the pumps to a functional status and are excluded from this contribution. A similar expression can be given for the interval test unavailability, $T = f \times t/720$, where f is the testing frequency as required by technical specifications.

2.8.2.2.4 Human Error Contribution, H - Young and Conradi [86] who participated in the Study identified that human error contributed to ESS unavailability in three major ways:

1. Operational errors such as premature or inadvertent shutdown of subsystems, erroneous switch operation, misinterpretation of procedures,
2. Testing errors whereby subsystems are exposed to loads or stresses beyond design limits, improper test equipment and improper test configurations.
3. Maintenance faults such as failure to return a system to operational readiness and miscalibration of sensor circuits.

In the case where procedures are repetitive or similar, the concept of coupling was used in quantifying human error. Four levels of coupling were used in the Study: No coupling (i.e., complete independence), loose coupling, tight coupling, and complete coupling (complete dependence).

As an example of coupling, consider the CSIS. During test of the CSIS, manual valves in both legs must be opened. If the valves are left

open after test, then enough water would be diverted to disable the entire CSIS in the event of a LOCA. It was estimated that the probability of leaving one valve open due to human error is 10^{-2}. If the actions of closing both valves after the test are assumed to be independent, then the probability of both valves being open due to human error is $(1 \times 10^{-2})(1 \times 10^{-2}) = 1 \times 10^{-4}$ as compared to 1×10^{-2} for complete dependence. The log-normal median between these two values results in the loosely coupled value of $\sqrt{(1 \times 10^{-2})(1 \times 10^{-4})} = 1 \times 10^{-3}$. The Study assumed the latter value of 1×10^{-3} to be valid in this case.*

In other cases, the Study assumed two human actions to be completely dependent. For example, procedures for operation action in realigning the suction of the low pressure injection pumps after LOCA were ambiguous; this lead to the assumption that two separate actions of manipulating switches to open V10 and V11 to be completed coupled. Related human actions that could simultaneously fail both redundant legs were referred to as the common mode contribution for system unavailability.

In some cases, a single human action that could disable an entire engineered safeguard system was identified. During maintenance of the LPIS, motor operated valves V9 and V10 are closed. If the operator forgets to open either V9 or V10, the entire LPIS is disabled. These two acts of omission represented 53% of the total calculated LPIS unavailability.

*Note that the concept of coupling introduces another method of quantificatively evaluating fault trees when basic events are statistically dependent.

2.8.2.2.5 System Unavailability, S - For one of two redundant legs, the leg unavailability due to hardware, test and maintenance is given by

$$S_L = Q + M + T.$$

For two redundant legs, A and B, the total system unavailability, S, is given by $S_A \cdot S_B$, i.e.,

$$S + Q_A \cdot Q_B + Q_A \cdot (M_B + T_B) + Q_B \cdot (M_A + T_A) + Q_{CM} + Q_{Singles} \quad (2.65)$$

where Q_{CM} is the unavailability due to human actions that are considered coupled, and $Q_{singles}$ are human and hardware failures that can disable the entire system.

Note that S does not include $(M_A + T_A) \cdot (M_B + T_B)$ since technical specifications prohibit maintenance of testing on two legs simultaneously when the reactor is at full power.

In expression (2.65), the terms

$$Q_{HDW} = Q_A \cdot Q_B + Q_{singles,\ hardware} \quad (2.66)$$

were called the hardware contribution; expression (2.67),

$$Q_{TM} = Q_A(M_B + T_B) + Q_B(M_A + T_A) + Q_{singles} \quad (2.67)$$

was referred to as the test and maintenance contribution, and $Q_{singles}$ refer to hardware or human failures that are related to test and maintenance action.

Calculations on the CSIS [70] show that the hardware contribution is dominated by doubles, i.e., $Q_{HDW} = Q_A \cdot Q_B = (1.8 \times 10^{-2})^2 = 3.2 \times 10^{-4}$

where Q_A (or Q_B) is, in turn, dominated by the independent event of the maintenance crew failing to open one CSIS manual valve after test, its probability given as 10^{-2}.

The test and maintenance contribution can be calculated by recalling that

$$M_A = M_B = 2.2 \times 10^{-3} \quad \text{Section 2.8.2.2.2}$$
$$T_A = T_B = 1.9 \times 10^{-3} \quad \text{Section 2.8.2.2.3,}$$

then (2.67) becomes (by symmetry)

$$Q_{TM} = 2(1.9 \times 10^{-3} + 2.2 \times 10^{-3})(1.8 \times 10^{-2}) = 1.5 \times 10^{-4}.$$

Now we consider the common mode contribution to CSIS unavailability. Recall that it is the consequence limiting control system, CLCS, that initiates CSIS operation. The study estimated that the probability of miscalibrating all sensors in the CLCS is 1×10^{-3}. Another common mode contribution mentioned previously is the case of leaving both manual valves closed after test (see Section 2.8.2.2.4); in this case, the common mode contribution is calculated to be

$$1 \times 10^{-3} - 1 \times 10^{-4} = 9 \times 10^{-4}.$$

The subtraction is needed since the independent actions of closing both manual valves separately is included in the hardware contribution.

The common mode contribution is computed to be

$$Q_{CM} = 1 \times 10^{-3} + .9 \times 10^{-3} = 1.9 \times 10^{-3}.$$

The probability that the CSIS is unavailable given a LOCA is then the

sum of the three contributions,

$$S_{CSIS}|LOCA = Q_{HDS} + Q_{TM} + Q_{CM}$$

$$= 3.2 \times 10^{-4} + 1.5 \times 10^{-4} + 1.9 \times 10^{-3}$$

$$= 2.4 \times 10^{-3}.$$

Vesely's [80] compilation of the relative contribution of Q_{HDW}, Q_{TM}, and Q_{CM} to system unavailability for various ESS systems considered in the Study is given in Table 2.5.

TABLE 2-5

Contributions to System Unavailability for
Various Engineered Safeguard Systems

SYSTEM	HARDWARE	TEST & MAINTENANCE	HUMAN
Low pressure recirculation system (LPR)	14%		47%
Sodium hydroxide system (NaOH)		75%	18%
Safety injection control system (SICS)	51%	38%	
Low pressure injection system (LPIS)	15%	20%	53%
Consequence limiting control system (CLCS)			91%
Containment leakage (CL)	65%		
Reactor protection (RP)	44%	33%	

The contributions do not add to 100% because there are other failure causes, such as environment-caused failures, failures due to combination of human errors and hardware failures, etc., not listed.

2.8.2.3 Confidence Limits on System Unavailability - In general, it was noted there was a wide range in the data collected. To account for this variability, failure rates, maintenance duration tests, test duration times, and maintenance intervals were assumed to be random variables with log normal distributions (a Bayesian approach in which the uncertainties in the above quantities are described by log-normal prior distributions). Using Monte Carlo simulation with a thousand trials for each system, the median and the 90% confidence levels for system unavailability were estimated. These results are plotted in Fig. 2.16 for the various engineered safeguard systems given in Table 2-5. The error bars in Figure 2.16 represent uncertainties in system failure probabilities that are due to uncertainties in the input data.

2.8.3 Containment Failure Modes - The magnitude of the radiological release is determined by the containment failure mode and the time at which failure occurs. Because of uncertainties concerning the accident phenomenology, containment failure mode probabilities were obtained by best engineering judgment. Wide error bands are associated with these probabilities.

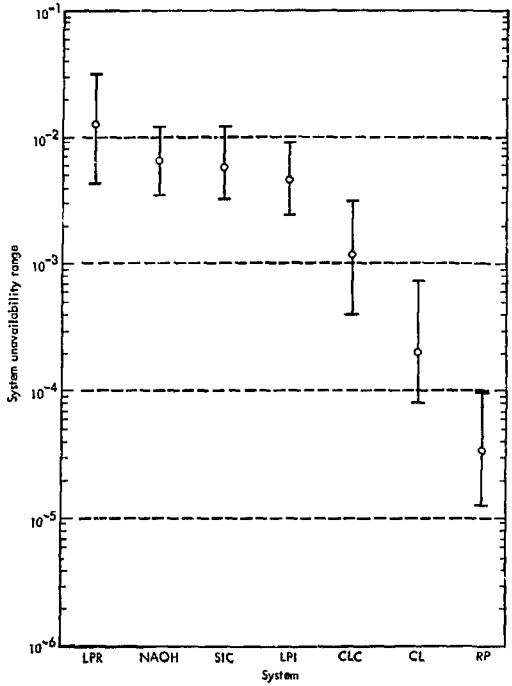


FIG. 2.16 Characteristic System Results

CHAPTER THREE

MEASURES OF IMPORTANCE OF EVENTS AND CUT SETS IN FAULT TREES

3.1 Introduction

In this chapter, we present a survey of the available methods that quantitatively rank basic events and cut sets according to their importance. Such a ranking permits identification of events and cut sets that significantly contribute to the occurrence of the top event. Time-dependent behavior of each method is shown, assuming proportional hazard rates and unrepairable components. Methods are presented to compute the importance of events for which repair is permitted. The practical application of importance measures for upgrading system designs, locating diagnostic sensors, and for generating checklists for system diagnosis is considered in Chapters Four and Five.

In Chapter One, we defined a system as an orderly arrangement of components that performs some task or function. It is clear by the arrangement of these components that some are more critical with respect to the functioning of the system than others. For example, when considering reliability, a component placed in series with the system generally plays a much more important role than that same component placed in parallel with the system. Another factor determining the importance of a component in a system is the reliability of the component, i.e., the probability that the component is working successfully. Measuring the relative importance of components may

- Identify components that merit additional research and development, thereby improving the overall reliability at minimum cost

or effort

- Suggest the most efficient way to diagnose system failure by generating a repair checklist for an operator to follow.

The fault tree is the most generalized Boolean model capable of identifying those basic causes that can contribute to system failure. These basic causes or events include environmental conditions, human error, and normal events (events that are expected to occur during the life of the system) as well as hardware failures. If the relative failure rates of the basic events are known, the fault tree can be quantitatively evaluated to assess their importance.

Several probabilistic methods can be used to compute the importance of basic events in the fault tree. All the methods assess the importance of basic events by a numerical ranking. The probabilistic interpretation describing the relationship of the occurrence of a basic event to the occurrence of the top event is different in each case.

One purpose of this chapter is to give the reader physical insight into the concepts of probabilistic importance so that he may better understand their applications. The reader is referred to Barlow and Proschan [4] and Chatterjee [10] for a more mathematical presentation of probabilistic importance.

3.2 Probabilistic Expressions that Measure Importance

3.2.1 Assumptions in Quantitative Calculations - In this chapter, it is assumed that all basic events are statistically independent. Computing probabilistic importance when basic events are associated (see Section 2.4.4) is discussed by Chaterjee [10].

No generality in methodology is lost if we assume that basic events are statistically independent. Further, it is assumed (unless otherwise indicated) that all basic events have an infinite fault duration time (i.e., in the case of components, repair is not permitted). Hence, g is only a function of $\underline{F}(t)$, where g is defined in Section 2.6. It is shown later than the same methods apply in describing the importance of events with finite fault duration times.

3.2.2 Measures Describing System Behavior at One Point in Time - We now introduce three measures of importance computed in terms of $g(\underline{F}(t))$, a function that measures the age of the system at t and describes system behavior at one point in time. Later, we introduce measures of importance that describe system failure in terms of sequences of component failures that cause the system to fail in time. These measures are functions of the past behavior of the system while the three we introduce now are not.

3.2.2.1 Birnbaum's Measure of Importance - In 1969, Birnbaum [8] introduced the concept of importance for coherent systems. He defined the reliability importance of a component i as the rate at which system reliability improves as the reliability of component i improves. If we construct a fault tree where the top event is system failure and the basic events are component failures,* then Birnbaum's definition of component importance becomes

$$\frac{\partial g(\underline{F}(t))}{\partial F_i(t)} = g(1_i, \underline{F}(t)) - g(0_i, \underline{F}(t)) \stackrel{\text{def}}{\equiv} \Delta g_i(t). \qquad (3.1)$$

*At this point, it is convenient to denote basic events as component failures when describing methods that measure importance. Used in this context, event importance is synonymous with coponent importance.

Some mathematical properties of $\Delta g_i(t)$ are

P1. $0 \leq \Delta g_i(t) \leq 1$.

P2. $\Delta g_i(t)$ does not depend upon $F_i(t)$ since $g(\underline{F}(t))$ is a linear function of $F_i(t)$ and basic events are statistically independent.

P3. If the set M with structure function χ is a module of $\Psi(\underline{Y})$, let $h(\underline{F}(t)) = E[\chi(\underline{Y}^M(t))]$ then

$$\Delta g_i(t) = \frac{\partial g[\underline{F}(t)), h(\underline{F}(t)]}{\partial h(\underline{F}(t))} \frac{\partial h(\underline{F}(t))}{\partial F_i(t)}.$$

In other words, if we know that a component is contained in a module, to compute the importance of the component to the system, we take the product of (1) the importance of the module to the system, and (2) the importance of the component to the module.

P4. For structures where at least two min cut sets do not overlap

$$\lim_{t \to \infty} \frac{\partial g(\underline{F}(t))}{\partial F_i(t)} = 0.$$

Birnbaum's definition of importance is also known by two other names, (1) marginal importance, and (2) the partial derivative.

Stated in other terms, $\Delta g_i(t)$ is the probability that the system is in a state at time t in which the functioning of component i is critical: the system functions when i functions, the system fails when i fails. The failure of i is critical at time t when $\Psi(1_i, Y(t)) = \Psi(0_i, Y(t)) = 1$.

On interest might be the total number of vector states for which a component is critical. If we fix the state of a component in the system, we are left with 2^{n-1} states, where n equals the number of components. In the above expression, if we let $F_j(t) = 1/2$ for all $j \neq i$, then the number of states in which component i is critical, denoted by B_i, is

$$B_i = 2^{n-1} \{g(1_i, 1/2) - g(0_i, 1/2)\}. \qquad (3.2)$$

Birnbaum calls B_i the structural importance of component i. [8]

For example, the fault tree shown in Fig. 3.1 exhibits three states in which the failure of 1 is critical.

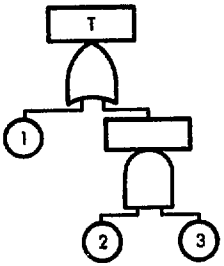


FIG. 3.1 Fault Tree with AND and OR Gates

(1) $Y_2 = 0$ and $Y_3 = 0$

(2) $Y_2 = 1$ and $Y_3 = 0$

(3) $Y_2 = 0$ and $Y_3 = 1$.

The number of critical cut sets for component 1 can be determined by using Equation 3.2. The structure function $\Psi(\underline{Y})$ is given by

$$\Psi(\underline{Y}) = Y_1 \amalg Y_2 \cdot Y_3$$

$$= 1 - (1 - Y_1)(1 - Y_2 \cdot Y_3)$$

for i = 1, $B_1 = 2^{3-1} \cdot \{1 - (1 - 3/4)\} = 3$ as verified above.

The sets of event {1}, {1,2}, and {1,3} are known as critical cut sets for component 1. The set {2,3} is a critical cut set for components 2 and 3. Note that a minimal cut set containing i is always a critical cut set for i. We see for a set of events to be a critical cut set for event i, each cut set contained in this set must contain the event i.

3.2.2.2 Criticality Importance - Birnbaum's definition of importance is a conditional probability in the sense that the state of the i^{th} component is fixed. The probability that the system is in a state at time t in which component i is critical and that component i has failed by time t is

$$\{g(1_i, \underline{F}(t)) - g(0_i, \underline{F}(t))\} F_i(t).$$

If we make this conditional to system failure by time t, then the above expression becomes

$$\frac{\{g(1_i, \underline{F}(t)) - g(0_i, \underline{F}(t))\} F_i(t)}{g(\underline{F}(t))} \stackrel{\text{def}}{=} I_i^{CR}(t). \qquad (3.3)$$

The above expression is defined as the criticality importance of component i. Note that $I_i^{CR}(t)$ is a function of $F_i(t)$ while $\Delta g_i(t)$ is not.

3.2.2.3 Vesely-Fussell Definition of Importance - It is possible that when system failure is observed, two or more cut sets could have failed. In this case, restoring a failed component to a working state does not necessarily mean that the system is restored to a working state. In other words, it is possible that a failure of a component can be contributing to system failure without being critical. Component i is contributing to system failure if a cut set containing i has failed; in terms of coherent structure theory notation

$$\psi_K^i(\underline{Y}(t)) = \prod_{j=1}^{N_K^i} \prod_{\substack{\ell \in K_j \\ i \in K_j}} Y_\ell(t) = 1,$$

where $\prod_{i \in K_j}^{\ell \in K_j}$ means that the index ℓ includes all basic events in cut set K_j,

where K_j contains the basic event i.

N_K^i = number of cut sets that contain basic event i;

$\psi_K^i(\underline{Y}(t))$ = Boolean indicator variable for the union of all cut sets that contain basic event i.

The probability that component i is contributing to system failure, $[\psi_K^i(\underline{Y}(t)) = 1]$, is denoted as $g_i(\underline{F}(t))$. The probability that component i is contributing to system failure, given that the system has failed by time t, is given by

$$\frac{g_i(\underline{F}(t))}{g(\underline{F}(t))} \overset{\text{def}}{=} I_i^{VF}(t). \tag{3.4}$$

This concept of importance was introduced by Vesely [78] and also Fussell [25], who later described it. Chatterjee calls $I_i^{VF}(t)$, the diagnostic importance of i.

We list the properties of the Vesely-Fussell definition of importance given by Chatterjee in reference [10].

P1. $0 \leq I_i^{VF}(t) \leq 1$.

P2. Let $Q_K(t) = \prod_{j \in K} F_j(t)$, then

$$I_i^{VF}(t) \leq \sum_{\substack{j \\ i \in K_j}} Q_{K_j}(t) \Big/ g(\underline{F}(t)).$$

Vesely and Narum [82] in their KITT computer program use the bound in P2 to approximate $I_i^{VF}(t)$. For large t, this may be a crude approximation. The IMPORTANCE computer code uses the min cut upper bound in computing $P[\Psi_K^i(\underline{Y}(t)) = 1]$ and is a much more accurate approximation in computing $I_i^{VF}(t)$ for large t.

P3. $I_i^{VF}(t)$ posses the same property as $\Delta g_i(t)$ for module decomposition, i.e.,

$$I_i^{FV}(t) = \frac{g_M(\underline{F}(t))}{g(\underline{F}(t))} \cdot \frac{h_i(\underline{F}(t))}{h(F(t))}$$

where $g_M(\underline{F}(t)) = P[\Psi_K^M(\underline{Y}(t)) = 1]$, $h_i(F(t)) = P[x_K^i(\underline{Y}(t)) = 1]$,

where χ is the structure function for the module M of Ψ, the structure function for the top event.

P4. $\lim_{t \to \infty} I_i^{FV}(t) = 1$

since all cut sets containing i eventually fail.

Note that if we substitute $g_i(\underline{F}(t))$ for $g(\underline{F}(t))$ in the definition of criticality importance, we obtain

$$\frac{\{g_i(1_i, \underline{F}(t)) - g_i(0_i, \underline{F}(t))\} F_i(t)}{g(\underline{F}(t))}.$$

Noting that

$$g_i(0_i, \underline{F}(t)) = 0$$
$$g_i(1_i, \underline{F}(t))F_i(t) = g_i(\underline{F}(t)),$$

we obtain the Vesely-Fussell definition of component importance

$$\frac{g_i(\underline{F}(t))}{g(\underline{F}(t))}.$$

Indeed, when component i is contributing to system failure, it is always critical to the structure $\psi_K^i(\underline{Y}(t))$.

3.2.3 Sequential Measures of Importance - The measures of importance presented thus far gives no information about the way system failure occurred. We now consider the way components fail sequentially in time to cause system failure. We first consider a measure of importance first given by Barlow and Proschan.

3.2.3.1 Barlow-Proschan Measure of Importance - Barlow and Proschan [4] examined components as they fail sequentially in time. They assume that if two or more components have a vanishingly small probability of occurring at the same instant, then one component must have caused the system to fail. The probability that event i causes the system to fail during a differential time interval of t', where $t' \leq t$, is

$$\{g(1_i, \underline{F}(t')) - g(0_i, \underline{F}(t'))\}\, dF_i(t').$$

Integrating between 0 and t

$$\int_0^t \{g(1_i, \underline{F}(t')) - g(0_i, \underline{F}(t'))\}\, dF_i(t') \qquad (3.5)$$

we get the probability that component i causes the system to fail in [0, t].

Barlow and Proschan [4] as well as Chatterjee [10] integrate (3.5) over [0, ∞]. However, there may be a dramatic difference in the ranking of components over time using expression (3.5); hence we shall retain the upper limit t, usually thought of as mission time.

It can be shown that [4]

$$\sum_{i=1}^{n} \int_0^t \{g(1_i, \underline{F}(t')) - g(0_i, \underline{F}(t'))\}\, dF_i(t') = g(\underline{F}(t)) \quad (3.6)$$

i.e., (3.6) is the probability that the system fails before t, where n is the number of components comprising the system. As shown in Section 2.6.1, expression (3.6) is simply the expected number of system failures in [0, t].

The conditional probability that a component i causes the system to *fail by the time t is then the Barlow-Proschan (B-P)* measure of importance

$$\frac{\int_0^t \{g(1_i, \underline{F}(t')) - g(0_i, \underline{F}(t'))\}\, dF_i(t')}{\sum_{i=1}^{n} \int_0^t \{g(1_i, \underline{F}(t')) - g(0_i, \underline{F}(t'))\} dF_i(t')} \stackrel{\text{def}}{=} I_i^{BP}(t). \quad (3.7)$$

The sum of all component importances in Barlow's measure of importance is unity. Essentially, B-P's measure of importance of a component i is the probability of the system failing because a critical cut set containing i fails, with component i failing last.

Barlow and Proschan define the structural importance of component i as the probability that component i causes the system to fail, assuming that all component failure probabilities are equal. Then they integrate from time t = 0 to t = ∞, or equivalently from q = 0 to q = 1

$$\int_0^1 [g(1_i, \underline{q}) - g(0_i, \underline{q})]dq, \tag{3.8}$$

where q = F(t). Again, it may be more appropriate in integrating (3.8) over [0, t] in assessing structural importance as given in (3.8).

We state two properties given by Barlow and Proschan concerning the evaluation of $I_i^{BP}(t)$ by modular decomposition

P1. $$I_i^{BP}(t) = \int_0^t [g(1^M, \underline{F}(t)) - g(0^M, \underline{F}(t))] \, [h(1_i, \underline{F}(t)) - h(0_i, \underline{F}(t))]dF_i(t).$$

P2. $$I_M^{BP}(t) = \sum_{i \in M} I_i^{BP}(t)$$

where g, h, and M have the same meaning as in Section 3.2.2.3.

3.2.3.2 Sequential Contributory Importance - It might be interesting to assess the role of the failure of a component i when another component, say j, causes the system to fail. The failure of i is a factor in this case only if i and j are contained in at least one min cut set. The probability that component i is contributing to system failure when j causes the system to fail is

$$\frac{\int_0^t \{g(1_i, 1_j, \underline{F}(t')) - g(1_i, 0_j, \underline{F}(t'))\} F_i(t') dF_j(t')}{g(\underline{F}(t))} \qquad (3.9)$$

and, in general, the probability that component i is contributing to system failure when another component causes the system to fail is

$$\frac{\sum_{j \neq j} \int_0^t \{g(1_i, 1_j, \underline{F}(t')) - g(1_i, 0_j, \underline{F}(t'))\} F_i(t') dF_j(t')}{g(\underline{F}(t))}$$

$$= I_i^{SC}(t), \qquad (3.10)$$

where the sum over j is to include only those components that appear in at least one min cut set with component i. Expression (3.10), $I_i^{SC}(t)$, shall be called the sequential contributory importance of component i.

3.3 Assumption of Proportional Hazards

To compare the time-dependent behavior of each method that measures importance, we must know the basic event probabilities, $F_i(t)$; this implies knowledge of $\lambda_i(t)$. In many cases, the failure rates are known to a poor degree of accuracy. However, using engineering judgment based on experience, the relative failure rates may be more accurately known. Furthermore, if we assume that all the failure rates exhibit the same time-dependent behavior (assumption of proportional hazards) then $F_i(t)$ may be written as

$$F_i(t) = 1 - e^{-R(t)\lambda_i}$$

for $i = 1, 2, \ldots, n$; where $R(t)$ is the common hazard and

$$\lambda_i = \frac{\int_0^t \lambda_i(t')dt'}{R(t)} .$$

If we arbitrarily select a reference, λ_j from $\underline{\lambda}$,* we may express $F_i(t)$ in terms of $F_j(t)$:

$$F_i(t) = 1 - (1 - F_j(t))^{\lambda_i/\lambda_j}.$$

Letting $\alpha_i = \lambda_i/\lambda_j$ and $q(t) = F_j(t)$, $F_i(t)$ becomes

$$F_i(t) = 1 - (1 - q(t))^{\alpha_i}, \tag{3.11}$$

where α_i is defined as the proportional hazard for basic event i.

3.4 Time-Dependent Behavior of Importance Measures

Under the assumption of proportional hazards, the results of each method can either be plotted as a function of $q(t)$ and $\underline{\alpha}$ or as a function of $g(\underline{F}(t))$ and $\underline{\alpha}$ since $g(\underline{F}(t))$ is a function of $q(t)$ (and $\underline{\alpha}$). We chose three systems to compare each measure of importance. These are referred to as systems A-3, B-3, and C-3.

System A-3 is a parallel system with components 1 and 2. The fault tree is shown in Fig. 3.2a and a corresponding reliability network diagram is shown in Fig. 3.2b. We assume a proportional hazard rate of 0.01 for component 1 and 1 for component 2; i.e., $\alpha_1 = 0.01$, and $\alpha_2 = 1$. In this case, $F_1(t) = 1 - (1 - q(t))^{0.01}$, $F_2(t) = q(t)$, and $g(\underline{F}(t)) = q(t) - q(t)(1 - q(t))^{.01}$. Five measures of importance are

*Where $\underline{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ and n is the number of basic events in the fault tree.

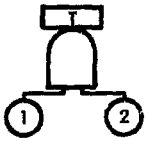


FIG. 3.2 System A-3 Fault Tree; the Structure Function is $\psi(Y_1, Y_2) = Y_1 \cdot Y_2$.

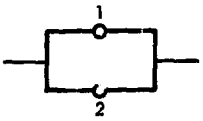


FIG. 3.2b Reliability Network Diagram

plotted vs $g(\underline{F}(t))$ in Fig. 3.3. They include Birnbaum, expression (3.1); criticality, expression (3.3); Vesely-Fussell, expression (3.4); Barlow-Proschan, expression (3.7) and the upgrading function

$$\frac{\alpha_i}{g(q(t), \underline{\alpha})} \cdot \frac{\partial g(q(t)), \underline{\alpha})}{\partial \alpha_i} .$$

The significance of the upgrading function is discussed in Chapter Four when upgrading of systems is considered.

We note in Fig. 3.3 that the probability that each component either contributes to or is critical to system failure is unity in each case. Barlow's and Birnbaum's definition of importance indicates that component 1 is more important. In a parallel system, the system fails when the last component fails; in this case, component 1 is more likely to fail last and cause the system to fail. Birnbaum's measure of importance tells us that System A is most likely to be in a state in which the failure of component 1 is critical.

System B-3 is a series system of two components 1 and 2. We can assume the same proportional hazard rate as in System A-3. In this case, $g(\underline{F}(t)) = 1 - (1 - q(t))^{1.01}$. The fault tree and corresponding network diagram are shown in Fig. 3.4

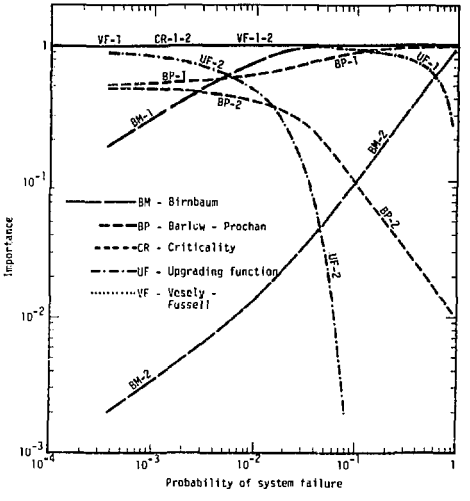


FIG. 3.3 Plots of Importance Measures for System A-3

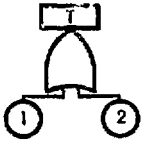


FIG. 3.4a System B-3 Fault Tree; the Structure Function is $\Psi(Y_1, Y_2) = 1 - (1-Y_1)\cdot(1-Y_2)$

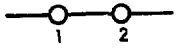


FIG. 3.4b Reliability Network Diagram

The plots in Fig. 3.5 show that component 2 is more important than component 1 in all cases. This is to be expected since component 2 has a failure rate 100 times greater than component 1 and a series system fails when the first component fails.

System C-3 is a series-parallel system. Component 1 is in series with a parallel structure of two components, 2 and 3. The fault tree and corresponding network diagram are shown in Fig. 3.6. For this example, it is assumed that $\alpha_1 = 0.1$ and $\alpha_2 = \alpha_3 = 1$. Figure 3.7 indicates that for small $g(\underline{F}(t))$ or small times t, component 1 is more important.* For large $g(\underline{F}(t))$ ($\gtrsim 0.05$) or large t, components 2 and 3 are more important. There is disagreement, however, as to which value of $g(\underline{F}(t))$ would make components 2 and 3 more important than component 1.

It can be seen from Figs. 3.3, 3.5, and 3.7 that each method produces a different time-dependent behavior; i.e., there is disagreement in the assessment of importances. The analyst should carefully define the probabilistic information he seeks regarding his system and then apply the appropriate measure of importance.

3.5 Cut Set Importance

Definitions of cut set importance are described by analogy to methods that determine component importance.

In the Vesely-Fussell definition, the importance of a cut set K_j is the probability that cut set K_j is contributing to system failure. It is given by

*Again, the value t can be thought of as mission time.

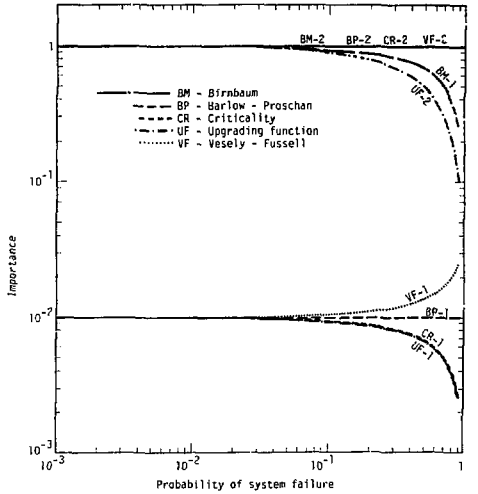


FIG. 3.5 Plots of Importance Measures for System B-3

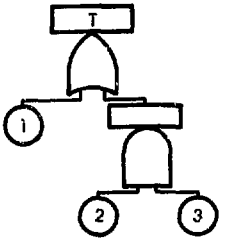


FIG. 3.6a System C-3 Fault Tree; the Structure Function is $\Psi(\underline{Y}) = 1-(1-Y_1)\cdot(1-Y_2\cdot Y_3)$.

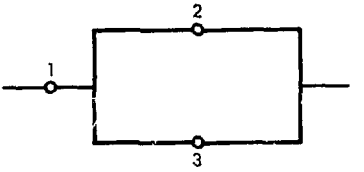


FIG. 3.6b Reliability Network Diagram

$$\frac{\prod_{i \in K_j} F_k(t)}{g(\underline{F}(t))} \quad . \tag{3.13}$$

The Barlow-Proschan definition of the importance of a cut set K_j is the probability that a cut set K_j causes the system to fail. For a cut set K_j to have caused the system to fail, a basic event contained in the cut set must have caused the system to fail and all other events in the cut set must have failed prior to the event that caused the system to fail.

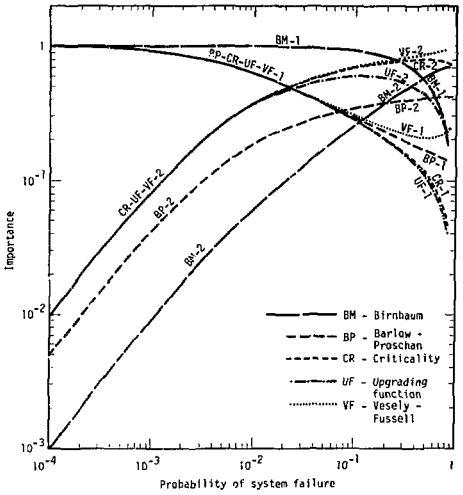


FIG. 3.7 Plots of Importance Measures for System C-3

B-P's measure of importance of a cut set K_j is

$$\frac{\sum_{i \in K_j} \int_0^t [g(\underline{1}^{K_j}, \underline{F}(t)) - g(0_i, \underline{1}^{K_j - \{i\}}, \underline{F}(t))] \prod_{\substack{\ell \neq i \\ \ell \in K_j}} F_\ell(t) dF_i(t)}{g(\underline{F}(t))},$$

where $\underline{1}^{K_j}$ means that Y_i is equal to 1 for each basic event i contained in cut set K_j. Since $g(\underline{1}^{K_j}, \underline{F}(t)) = 1$, the above expression becomes

$$\frac{\sum_{i \in K_j} \int_0^t [1 - g(0_i, \underline{1}^{K_j - \{i\}}, \underline{F}(t))] \prod_{\substack{\ell \neq i \\ \ell \in K_j}} F_\ell(t) dF_i(t)}{g(\underline{F}(t))}. \qquad (3.14)$$

Vesely-Fussell's definition of cut set importance always assigns more importance to a cut set of a lower order than a cut set of a higher order when basic event probabilities are equal. This is not always true, however, with B-P's measure of importance. As an example, consider a 10 component system with min cut sets given by

$K_1 = \{1,2,3,4\}$	$K_6 = \{5,7,8\}$	$K_{11} = \{5,9,10\}$	$K_{16} = \{6,8,10\}$
$K_2 = \{5,6,7\}$	$K_7 = \{5,7,9\}$	$K_{12} = \{6,7,8\}$	$K_{17} = \{6,9,10\}$
$K_3 = \{5,6,8\}$	$K_8 = \{5,7,10\}$	$K_{13} = \{6,7,9\}$	$K_{18} = \{7,8,9\}$
$K_4 = \{5,6,9\}$	$K_9 = \{5,8,9\}$	$K_{14} = \{6,7,10\}$	$K_{19} = \{7,8,10\}$
$K_5 = \{5,6,10\}$	$K_{10} = \{5,8,10\}$	$K_{15} = \{6,8,9\}$	$K_{20} = \{7,9,10\}$
			$K_{21} = \{8,9,10\}$

No component of K_1 appear in other min cut sets. The remaining sets were obtained by taking all combinations of three components from the remaining six. For this system

$$g(\underline{F}(t)) = \text{Prob}\left[\bigcup_{i=1}^{21} \kappa_i = 1\right] = \text{Prob}\left[\kappa_1 \cup \bigcup_{i=2}^{21} \kappa_i = 1\right]$$

$$= 1 - (1 - \text{Prob}\,(\kappa_1 = 1))(1 - \text{Prob}\,(\bigcup_{i=2}^{21} \kappa_i = 1)),$$

where κ_i is the indicator variable for cut set K_i. Setting $q(t) = F_i(t)$ for all i, where i = 1 to 10

$$g(\underline{F}(t)) = 1 - (1 - q(t)^4)(1 - \sum_{j=3}^{6}\binom{6}{j}\,(1 - q(t))^{6-j}q(t)^j).$$

Substituting in expression (3.14), Barlow-Proschan's measure of importance for cut set K_1, I_{K_1} becomes

$$I_{K_1} = \frac{4\int_0^{q(t)} [1 - \sum_{j=3}^{6}\binom{6}{j}(1 - q')^j q'^{6-j}]q'^3 dq'}{g(\underline{F}(t))},$$

for cut set K_2

$$I_{K_2} = \frac{3\int_0^{q(t)} (1 - q'^4)(1 - q')^3\, q'^2 dq'}{g(\underline{F}(t))}.$$

The Vesely-Fussell definition of importance gives

$$I_{K_1} = \frac{q(t)^4}{g(\underline{F}(t))}, \quad I_{K_2} = \frac{q(t)^3}{g(\underline{F}(t))}.$$

In Fig. 3.8, the importances of cut sets K_1 and K_2 are plotted as a function of $g(\underline{F}(t))$. Cut set K_2 always has a greater probability of contributing to system failure than cut set K_1. However, for

$g(\underline{F}(t)) \gtrsim 0.64$, cut set K_1 has a greater probability of causing the system to fail. If the basic events contained in K_2 were not replicated in other cut sets, then K_2 would always have a higher failure probability of causing the system to fail in K_1. In general, when no replication of events occur, a lower order cut set is always more important than a higher order cut set when basic event probabilities are equal.

3.6 Importance of Components when Repair is Permitted

3.6.1 Rate of Breakdown at Steady State - Each of the methods previously described can also assess the importance of components when repair is permitted. In every importance expression except Barlow and Proschan's, the limiting unavailability, $\bar{A}_i$, can be substituted for $F_i(t)$ without any change in probabilistic meaning.

To motivate B-P's definition of component importance when repair is permitted, consider an unrepairable system that has failed at some specified time t. If component i has distribution F_i with density f_i ($i = 1, 2, \ldots, n$), then the probability that i caused system failure (given that the system failed precisely at time t) is

$$\frac{[g(1_i, \underline{F}(t)) - g(0_i, \underline{F}(t))]\, f_i(t)dt}{\sum_{j=1}^{n} [g(1_j, \underline{F}(t)) - g(0_j, \underline{F}(t))]\, f_j(t)dt}. \tag{3.15}$$

As described in Section 2.5.3, the process of repairing a failed component is called an alternating renewal process. In this case, the component alternates between two states, an upstate and a downstate. The probability that a failure occurs about some differential time interval is $w_{f,i}(t)dt$, called the renewal failure density. $w_{f,i}(t)$ is

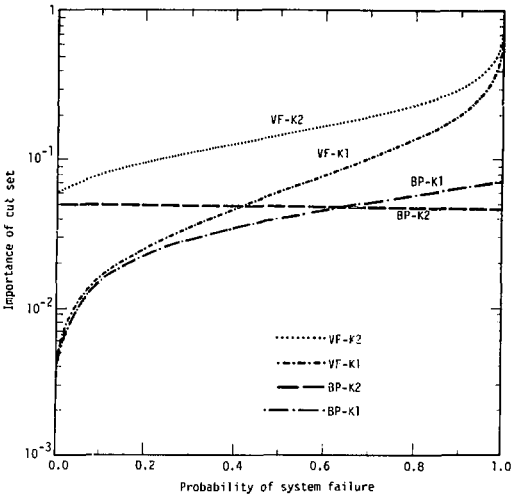


FIG. 3.8 Plots of Cut Set Importance

analogous to $f_i(t)$ in the nonrepairable case. The probability that a component is down at time t is $\bar{A}_i(t)$, called the unavailability of component i at time t (analogous to $F_i(t)$). The probability that component i caused system failure is

$$\frac{[g(1_i, \underline{\bar{A}}(t)) - g(0_i, \underline{\bar{A}}(t))]\, w_{f,i}(t)dt}{\sum_{i=1}^{n}[\, g(1_i, \underline{\bar{A}}(t)) - g(0_i, \underline{\bar{A}}(t))]w_{f,i}(t)dt},$$

where

μ_i = mean time to failure for component i

τ_i = mean time to repair for component i

$$\lim_{t\to\infty} \bar{A}_i(t) = \frac{\tau_i}{\mu_i+\tau_i}.$$

Letting $t\to\infty$, we obtain the stationary probability that component i causes system failure

$$\frac{[g(1_i, \underline{\bar{A}}) - g(0_i, \underline{\bar{A}})]/(\mu_i + \tau_i)}{\sum_{j=1}^{n}[g(1_j, \underline{\bar{A}}) - g(0_j, \underline{\bar{A}})]/(\mu_j + \tau_j)} \overset{\text{def}}{=} I_i^{BP,SS}. \qquad (3.16)$$

As the following discussion shows, the result is reasonable on physical grounds. $\mu_i + \tau_i$ is the average amount of time between failures for component i; i.e., the average length of time for a renewal cycle (see Section 2.5.3.1.4). $1/(\mu_i + \tau_i)$ is the average rate at which component i fails in the steady state, i.e., $w_{f,i}(\infty) = 1/(\mu_i + \tau_i)$. At large times, the system failure probability is time-invariant since the probability that each component fails is time-invariant.

3.6.2 Rate of First Failure Predicted by T* Method - The T* Method described in Section 2.6.2.4.5 provides a direct way of determining the probability that a component causes the system to fail for the first time in [0, t] when repair is allowed. We can assess the importance of a component in terms of the T* method as

$$I_i^{T*}(t) = \frac{g_{s,i}(t)}{\sum_j g_{s,j}(t)} \qquad (3.17)$$

where $g_{s,i}(t)$ is given by expression (2.45). Expression (3.17) is the probability that component i causes system failure for the first time in [0, t].

3.6.3 Rate of First Failure Predicted by Steady-State Upper Bound - If the distribution of time to first failure is approximated by

$$F_{SS}(t) = 1 - \prod_{i=1}^{n} [1 - \Delta g_i]^{\frac{t}{\mu_i + \tau_i}},$$

expression (2.34), an expression analogous to (3.17) can be generated by integrating (2.38) over [0, t] and conditioning on the first system failure in [0, t]. The result is

$$I_i^{SS} = \frac{\frac{\ln[1-\Delta g_i]}{\mu_i+\tau_i}}{\sum_{j=1}^{n} \frac{\ln[1-\Delta g_j]}{\mu_j+\tau_j}} \qquad (3.18)$$

where $\Delta g_i = g(1_i, \underline{A}) - g(0_i, \underline{A})$.
Notice that I_i^{SS} does not depend on time.

We choose system 2-C, Section 2.6.2.4.6, to compute the importance of each component in the system by expressions (3.16); $I_i^{T*}(t)$, expression (3.17) and I_i^{SS}, expression (3.18). The results are given in Table 3-1.

TABLE 3-1

Listing of Component Importances for System 2-C

Component	$I_i^{T*}(.01\mu)$	$I_i^{T*}(.1\mu)$	$I_i^{T*}(1\mu)$	$I_i^{T*}(10\mu)$	I_i^{SS}	$I_i^{BP,SS}$
1, 2 or 3	.2220	.2210	.2201	2200	.2199	.2196
4	.3339	.3370	.3398	.3399	.3403	.3411

We see in Table 3-1, that there is close agreement between $I_i^{I^*}(t)$ and $I_i^{BP,SS}$. We see that in this example the rate in which component i causes the first system failure is very nearly the rate it causes system failure in the steady state.

3.7 IMPORTANCE Computer Code

A computer code called IMPORTANCE was written and is described in Appendix A. It requires as input the minimal cut sets; the failure rates and fault duration times of all basic events. The failure and repair distributions are assumed to be exponential. There are many options to the code concerning the input. The code computes as output the following measures of basic event importance, (1) Birnbaum, (2) Criticality, (3) Upgrading Function, (4) Vesely-Fussell, (5) Barlow-Proschan, (6) Sequential Contributory and two measures of cut set importance, (1) Barlow-Proschan and (2) Vesely-Fussell. The code will be available from the Argonne Code Center, Argonne National Laboratory.

3.8 Summary of Importance Measures

As a summary, we list in Table 3-2 all the measures of importance given in this chapter and describe briefly their probabilistic meaning. In this table the notation of Section 2.6 is adopted.

$$E[Y_i(t)] = q_i(t) = \begin{cases} F_i(t) & \text{if basic event i has an infinite fault duration time} \\ \bar{A}_i(t) & \text{if basic event i has a finite fault duration time (its ON availability)} \end{cases}$$

$$E[\Psi(\underline{Y}(t))] = g\,(\underline{q}(t)) = g\,(\underline{F}(t))$$

where $\Psi(\underline{Y}(t))$ is the indicator variable for the top event

$$w_{f,i}(t) = \begin{cases} f_i(t) & \text{(the density) if basic event i has an infinite fault duration time} \\ w_f(t) & \text{(the failure density in renewal theory) if basic event i has a finite fault duration time} \end{cases}$$

$g_i(\underline{q}(t))$ is the probability that a min cut set containing i is failed at time t, $g_{s,i}(t)$ is given by expression (2.45), and $\Delta g_i = g(1_i, \underline{A}) - g(0_i, \underline{A})$.

TABLE 3.2 Summary of Importance Measures

IMPORTANCE MEASURE	PROBABILISTIC EXPRESSION	MEANING
BIRNBAUM Basic Event Importance	$g(1_i, \underline{q}(t)) - g(0_i, \underline{q}(t))$	Probability that the system is in a state in which the occurrence of event i is critical.
CRITICALITY Basic Event Importance	$\frac{[g(1_i, \underline{q}(t)) - g(0_i, \underline{q}(t))]\, q_i(t)}{g(\underline{q}(t))}$	The probability that event i has occurred and is critical to system failure.*
UPGRADING FUNCTION Basic Event Importance	$\frac{\lambda_i(t)}{g(\underline{q}(t))} \frac{\partial g(\underline{q}(t))}{\partial \lambda_i(t)}$	Fractional reduction in the probability of the top event when $\lambda_i(t)$ is reduced fractionally.
VESELY-FUSSELL Basic Event Importance	$\frac{g_i(q(t))}{g(\underline{q}(t))}$	Probability that event i is contributing to system failure.*
BARLOW-PROSCHAN Basic Event Importance	$\int_0^t \{g(1_i, \underline{q}(t)) - g(0_i, \underline{q}(t))\}\, w_{f,i}(t)\, dt$	Expected number of failures caused by basic event i in $[0, t]$.
CONTRIBUTORY SEQUENTIAL Basic Event Importance	$\sum_{\substack{j \\ i \neq j}} \int_0^t \{g(1_i, 1_j, \underline{q}(t)) - g(1_j, 0_i, \underline{q}(t))\}\, q_i(t) dw_{f,j}(t)$ $i \& j \in K_\ell$ for some ℓ	The expected number of system failures in $[0,t]$ caused by min cut sets that contain basic event i with basic event j occurring prior to system failure.

*Given that system failure has occurred

TABLE 3.2 Cont'd

IMPORTANCE MEASURE	PROBABILISTIC EXPRESSION	MEANING
STEADY-STATE BARLOW-PROSCHAN Measure of Basic Event Importance	$\dfrac{[g(1_i, \underline{\bar{A}}) - g(0_i, \underline{\bar{A}})]/(\mu_i+\tau_i)}{\sum_{i=1}^{n} [g(1_i, \underline{\bar{A}}) - g(0_i, \underline{\bar{A}})]/(\mu_i+\tau_i)}$	Probability that event i causes system failure in the steady state.†
FIRST FAILURE RATE OF BREAKDOWN, T* Basic Event Importance	$\dfrac{g_{s,i}(t)}{\sum_i g_{s,i}(t)}$	Probability that event i causes first system failure approximated by T* method.†
FIRST FAILURE RATE OF BREAKDOWN, SS Upper Bound, Basic Event Importance	$\dfrac{\ln[1-\Delta g_i]/(\mu_i+\tau_i)}{\sum_{i=1}^{n} \ln[1-\Delta g_i]/(\mu_i+\tau_i)}$	Probability that event i causes first system failure approximated by steady-state upper bound method.†
VESELY-FUSSELL Cut Set Importance	$\dfrac{\prod_{i\in K_\ell} q_i(t)}{g(\underline{q}(t))}$	Probability that min cut set K_ℓ is contributing to system failure.*
BARLOW-PROSCHAN Cut Set Importance	$\sum_{i\in K_i} \int_0^t [1 - g(0_i, \underline{1}, \underline{q}(t))] \prod_{\substack{j\neq i \\ j\in K_i}} q_j(t)\, dw_{f,i}(t)$	Expected number of system failures caused by min cut set K_i.

*Given that system failure has occurred
†Maintained system

CHAPTER FOUR

APPLICATION OF PROBABILISTIC IMPORTANCE TO SYSTEM DESIGN

In this chapter we apply the concept of probabilistic importance in before-the-fact investigation. We use fault trees as a design tool in upgrading system designs to improve their safety or reliability. We also show how the concept of probabilistic importance can be used to determine the optimal location for sensors in a system.

4.1 Upgrading System Designs

It is common during the design stages of the system to assume that all components are unrepairable. If the importance measures are not sensitive functions of time, then the importance of each event can be assessed with knowledge of the proportional hazards only. This means that systems can be upgraded on the basis of quantitative information that is relative rather than absolute in nature.

4.1.1 Estimating the Proportional Hazard - The concept of proportional hazards is discussed in Section 3.3. On the basis of the discussion given in Section 1.9.6.1, we may assign proportional hazard rates to the following types of events given in Table 4-1, where the hazard rates given below are on a per-demand or per-cycle basis. The adjustment of these hazard rates to an hourly hazard rate depends upon the system operating characteristics in time. On the basis of engineering judgment, an analyst may want to account for the system environment or operating conditions in the assignment of proportional hazards. He may simply do so on the basis of the K factors mentioned in Section 2.5.1.1.

TABLE 4-1

Proportional Hazards for Human Error

and Component Failures

Basic Event	Proportional Hazard, α
Human Failure Rates	100 - 1
Quasi static components	10^{-2} - 10^{-4}
Dynamic components	
Hydraulic	1 - 10^{-2}
Dynamic Components	1 - 10^{-3}

4.1.2 Improving System Designs - A goal of fault tree analysis is to identify weaknesses inherent to a system. The first step in fault tree evaluation is to visually inspect the fault tree to see if there are any first-order cut sets, i.e., any basic events that can individually induce system failure.

If any such events are identified as making an unacceptably high contribution to the top event, the system must be upgraded, i.e., the importance or the criticality of these events must be reduced. To reduce the probability of a component contributing to system failure, one can (1) incorporate parallel or standby redundancy in the system, (2) increase the reliability of the component, e.g., by derating it, (3) design to fail safe, (4) incorporate safety devices, (5) test a standby component more often,* and (6) provide alternate modes of operation.

*See Section 2.5.2.3 that discusses the optimum test interval that minimizes the unavailability of a component.

Human error can contribute throughout the system cycle. Errors during construction and maintenance can be eliminated by rigid quality control. Errors in operation can be reduced by administrative procedures or by automating the system. The effect of maintenance errors can be minimized by double checking or by monitoring critical components, e.g., the position of a manual valve.* If the analyst foresees any likely environmental or operational stresses, then components must be designed to withstand these stresses.

In the following four subsections when upgrading system designs are considered, we assume all components to be unrepairable.

4.1.3 Upgrading Function - It is the author's contention that Birnbaum's measure of importance,

$$\frac{\partial g(\underline{F}(t))}{\partial F_i(t)},$$

cannot be practically applied for upgrading reliable systems. For a given incremental reduction Δx in $F_i(t)$, Birnbaum identifies the event i that has the greatest effect in reducing $g(\underline{F}(t))$; i.e.,

$$\frac{\partial g(\underline{F}(t))}{\partial F_i(t)}$$

identifies the event i for which the quantity

$$g[F_i(t), \underline{F}(t)] - g[F_i(t) - \Delta x, \underline{F}(t)]$$

*If such procedures compensating for human error were incorporated into the engineered safeguard systems discussed in Section 2.8.2.2.4, the unavailability of these systems could, in some cases, have been reduced considerably.

is a maximum. Note that the above difference does not depend upon $F_i(t)$ because

$$\frac{\partial g(\underline{F}(t))}{\partial F_i(t)}$$

is not a function of $F_i(t)$ if basic events are statistically independent. Recall that $\frac{\partial g(\underline{F}(t))}{\partial F_i(t)} = g(1_i, \underline{F}(t)) - g(0_i, \underline{F}(t))$. For reliable systems $F_i(t)$ varies typically between 10^{-8} to 10^{-1} (where t can be thought of as mission time). Thus, subtracting a given increment Δx from each basic event failure probability is not a good test for system upgrade because of the smallness and variability of $F_i(t)$. Instead, we must make fractional or relative changes in $F_i(t)$. This can be done my making Δx a function of $F_i(t)$:

$$\Delta x = \gamma F_i(t),$$

where γ is any given constant between 0 and 1.* The expression

$$g[F_i(t), \underline{F}(t)] - g[F_i(t) - \gamma F_i(t), \underline{F}(t)]$$

identifies the event i that has the greatest effect in reducing $g(\underline{F}(t))$ when $F_i(t)$ is multiplied by a given constant $1 - \gamma$. In taking the limit as γ approaches 1 in the above expression, we identify the difference as a differential quantity. Dividing the above expression by $1 - \gamma$,

---

*A similar argument based on fractional rather than incremental changes can be found in Appendix III, Section 3.6.1 of WASH 1400. [71] The Study found that the spread in failure rate data varied by multiplicative factors rather than incremental factors. The common and natural distribution for describing data that can vary by multiplicative factors is the log-normal distribution. The normal distribution, on the other hand, is natural for describing data that can vary by additive or subtractive increments. On this same basis we claim that the upgrading function is more appropriate for improving system reliability than is Birnbaum's measure of importance.

multiplying by $F_i(t)/F_i(t)$ (unity) we can then take the limit as $\gamma \to 1^-$,

$$\lim_{\gamma \to 1^-} F_i(t) \frac{g\{F_1(t),\ldots,F_i(t),\ldots,F_n(t)\}-g\{F_1(t),\ldots,F_i(t),\ldots F_n(t)\}}{F_i(t)(1-\gamma)}$$

and identify the above quantity as being

$$F_i(t) \frac{\partial g(\underline{F}(t))}{\partial F_i(t)}.$$

Note that the above expression is a function of $F_i(t)$ whereas $\frac{\partial g(\underline{F}(t))}{\partial F_i(t)}$ is not.

It is because of this reason that Birnbaum's measure of importance can give significance to a relatively insignificant event. For example, we can hypothesize that lightning striking a missile can cause auto ignition of the propellant and in turn cause an inadvertent launch of a missile. We can estimate the probability of this event, denoted as event A, to be 10^{-9}/yr. Furthermore, we may guess that the probability of an inadvertent missile launch due to all causes other than lightning is 10^{-7}/yr.

Birnbaum's measure of importance estimates the importance of the event A to be

$$1 - (1 - (1 - 10^{-7})) = .9999999 \approx 1.$$

On the other hand, criticality importance estimates the importance to be

$$\frac{.9999999}{1.01 \times 10^{-7}} 10^{-9} \approx .01.$$

The quantity that is physically measurable is the failure rate $\lambda_i(t)$ as opposed to a failure probability of $F_i(t)$. Hence, it is more meaningful to upgrade a system according to the following expression:

$$\lambda_i(t) \frac{\partial g(\underline{\lambda}(t))}{\partial \lambda_i(t)}.$$

If the analyst assumes that the failure rates are proportional (assumption of proportional hazards, see Section 3.3), changes in $\lambda_i(t)$ are equivalent to changes in α_i. Fractional or relative changes in α_i change $g(\underline{\alpha}, q(t))$ incrementally at a rate*

$$\alpha_i \frac{\partial g(\underline{\alpha}, q(t))}{\partial \alpha_i}$$

or fractionally at a rate

$$\frac{\alpha_i}{g(\underline{\alpha}, q(t))} \cdot \frac{\partial g(\underline{\alpha}, q(t))}{\partial \alpha_i}.$$

The last two expressions give the same relative ranking. The advantage of using the latter expression is that it yields numbers much closer to unity. It shall be denoted as the upgrading function.

If we identify a component failure with hazard rate α_i^I as the event for which

$$\frac{\alpha_i}{g(\underline{\alpha}, q(t))} \cdot \frac{\partial g(\underline{\alpha}, q(t))}{\partial \alpha_i}$$

is maximum, we may wish to replace the component with a more reliable component with a hazard rate of α_i^F. If

$$\frac{\alpha_i}{g(\underline{\alpha}, q(t))} \cdot \frac{\partial g(\underline{\alpha}, q(t))}{\partial \alpha_i}$$

*Recall from Section 3.3 that $q(t) = F_j(t)$ where $F_j(t)$ is the reference cumulative failure distribution function.

remains the maximum for all α_i between α_i^F and α_i^I, then the optimal course of system upgrade has been chosen. However, if there is a value of α_j, $\alpha_j^F \leq \alpha_j < \alpha_j^I$, in which another event j has a greater value:

$$\frac{\alpha_j}{g(\underline{\alpha}, q(t))} \cdot \frac{\partial g(\underline{\alpha}, q(t))}{\partial \alpha_j} > \frac{\alpha_i}{g(\underline{\alpha}, q(t))} \cdot \frac{\partial g(\underline{\alpha}, q(t))}{\partial \alpha_i},$$

then the absolute value of

$$g(\alpha_i \ldots, \alpha_i^I, \ldots, \alpha_n, q(t)) - g(\alpha_i \ldots, \alpha_i^F, \ldots, \alpha_n, q(t)) \text{ vs}$$

$$g(\alpha_i \ldots, \alpha_j^I, \ldots, \alpha_n, q(t)) - g(\alpha_i \ldots, \alpha_j^F, \ldots, \alpha_n, q(t))$$

must be calculated to determine the optimal choice of system upgrade.

4.1.4 Upgrading Systems Under Cost Constraints - Designers or manufacturers are always faced with cost constraints. They know that extremely reliable components are generally very expensive. It is an engineering challenge to manufacture a product that is safe and reliable and still economically competitive.

A designer may be faced with a basic design of n components. Contract specifications might require (1) that he design a system with a failure probability of less than g_0 for the system mission length, and (2) that the cost of the system be less than $\$_0$. For each component i he has a selection of m_i models or types to choose where $m_i \geq 1$. There are a total of $\prod_{i=1}^{n} m_i$ component selections for the system. The failure rate for the j^{th} selection of the i^{th} system component is denoted as λ_{i_j}, the cost of this component is denoted as $\$_i^j$. For a particular

selection $\underline{j}$ of n components, the cost of the system is $\sum_{i=1}^{n} \$_i \underline{j}$ with probability of system failure $g(\underline{\lambda}^{\underline{j}}, t)$, assuming constant failure rates.

A computer algorithm can be devised that chooses the optimal selection of $\underline{j}$ in which $\sum_{i=1}^{n} \$_{i_j} \leq \$_0$ and $g(\lambda_i \underline{j}, t) \leq g_0$. The upgrading function

$$\frac{\lambda_i}{g(\underline{\lambda}, t)} \frac{g(\underline{\lambda}, t)}{\partial \lambda_i}$$

can be used in the manner described previously to identify the critical components whose reliability must be improved. In general, it is possible to have two or more system designs; in this case, the computer can choose for each system the most optimal choice of $\underline{j}$. Decisions then can be made as to the best design.

4.1.5 Other Measures of Importance Considered in Upgrading Systems- For reliable systems, the upgrading function,

$$\alpha_i \frac{g(\underline{\alpha}, q(t))}{\partial \alpha_i}$$

may be approximated by the criticality expression,

$$F_i(t) \frac{\partial g(\underline{F}(t))}{\partial F_i(t)} .$$

Recall that $F_i(t) = 1 - (1 - q(t))^{\lambda_i}$ and $q(t) = F_j(t) = 1 - e^{-R(t)\alpha_i}$. This implies that $F_i(t) = 1 - e^{[-R(t)\lambda_j]\alpha_i}$. For reliable systems $R(t)$ is a small quantity, and $F_i(t)$ may be approximated by $R(t)\lambda_j\alpha_i$. Since $R(T)\lambda_j$ is a constant with respect to α_i, $F_i(t)$ is proportional to α_i; hence, for reliable systems

$$\frac{F_i(t)}{g(\underline{F}(t))} \quad \frac{\partial g(\underline{F}(t))}{\partial F_i(t)} \cong \frac{\alpha_i}{g(\underline{\alpha},\ q(t))} \quad \frac{\partial g(\underline{\alpha},\ q(t))}{\partial \alpha_i}.$$

As shown in Figure 3.3, for a parallel system of two components, the criticality importance of components 1 and 2 is unity. The upgrading function for component 2 approaches unity as $g(\underline{F}(t))$ approaches 0.

The criticality expression for reliable systems in turn can be approximated by the Vesely-Fussell definition of importance. For reliable systems, the rare event approximation

$$g(\underline{F}(t)) = \sum_{j=1}^{N} \prod_{l \epsilon K_j} F_l(t)$$

is a good approximation for $g(\underline{F}(t))$. $g(\underline{F}(t))$ further may be written as

$$g(\underline{F}(t)) = \sum_{j} \prod_{\substack{i \notin K_j \\ l \epsilon K_j}} F_l(t) + \sum_{j} \prod_{\substack{i \epsilon K_j \\ l \epsilon K_j}} F_l(t)$$

substituting the above into the criticality importance expression

$$\frac{[g(1_i,\ \underline{F}(t)) - g(0_i,\ \underline{F}(t))]F_i(t)}{g(\underline{F}(t))}$$

we get

$$\frac{\left[\sum_{j} \prod_{\substack{i \epsilon K_j \\ l \epsilon K_j}} F_{\ell}(t)\right] F_i(t)}{\substack{Y_i(t)=1 \\ g(\underline{F}(t))}}$$

which is the Vesely-Fussell definition of importance.

It is the author's opinion that it is more meaningful to upgrade systems by event importance rather than by cut set importance. When replication of basic events occurs in cut sets, it is difficult to look at a cut set as a discrete entity.

4.1.6 Example of System Upgrade - In Appendix B, we show an example of how the upgrading function can be used for recommending design improvements and comparing competing designs. We assume proportional hazards and show how decisions concerning the adequacies of systems can be based on relative rather than absolute determinations.

4.2 FMECA as a Sensitivity Analysis

Jordan [45] has proposed a method of performing a sensitivity analysis in terms of failure modes and effects and criticality analysis (FMECA). Component failure modes with class III or IV hazard categories are placed on a critical items list. (Recall from Section 1.6 that class III and IV hazards have a critical effect on the system or personnel). Component failure modes on the critical items list are grouped according to their effect on the system. For example, we consider a chemical processing system consisting of reactant and product streams and a chemical reactor. It is necessary in this system to cool the reactant streams by a heat exchanger because the chemical reaction in the reactor is exothermic. Table 4-1 is a critical items list that shows three failure modes of the heat exchanger that have different effects on the system. In a similar manner, other component failure modes may be listed according to their effect on the system. Then for each system effect, Jordan ranks each component according to the product of (1) probability of occurrence, and (2) the probability that the failure mode

will produce the system effect when the failure mode occurs. Such a computation and ranking is referred to as a criticality analysis.

TABLE 4-1

Critical Items List

Component	Failure Mode	System Effect	Hazard Classification
Heat exchanger HX	Coolant flow too high, reactant temperature too low	Product concentration too low	III
Heat exchanger	Coolant leak from shell side to tube side of HX	Product stream contaminated	III
Heat exchanger	HX plugged coolant side	Reactant temperature too high, potential for explosion	IV

As shown in Table 4-1, a component may have many failure modes that have different effects on the system. To assess the overall importance of a component, Jordan sums over all failure mode probabilities in the criticality analysis involving the component. The advantage of Jordan's approach is simplicity. The disadvantage is that FMECA considers hardware failures only, i.e., it is not as general as FTA. FMECA is also inefficient in considering multiple failures, i.e., FMECA is primarily a single failure analysis. FTA, on the other hand, is well suited for analyzing complex systems on a functional basis and can describe multiple failures.

4.3 Optimal Sensor Location

We now consider locating sensors in a system according to the probabilistic importance of basic events and intermediate gate events in a fault tree. In Section 4.3.1, we consider monitoring components directly that have a high probability of being critical to system failure. Then, in Section 4.3.2, we consider locating sensors in a system that monitors the state of a subsystem. We detect a fault in a subsystem by its effect on the system, i.e., by the abnormal changes in the physical properties of the system. Such physical properties include flow rate, pressure, concentration, temperature, netron flux level, etc These subsystem abnormalities can usually be described by intermediate events at the major systems level in a fault tree (see Fig. 1.15). In this case, we use modular decomposition in calculating the importance of a gate event for the top event in order to determine the optimal sensor location. The designer is faced with one practical constraint when locating these sensors in the system -- the response time of the system to a subsystem or component fault must be greater than the time required to detect and rectify the fault if system failure is to be prevented. In Chapter Five we consider the time response of the system to various types of fault conditions.

4.3.1 Preventive Sensors - In a truly redundant system, no single component failure can cause the system to fail. In these systems (assuming failures are statistically independent) at least one component must fail prior to system failure. System failure can be prevented by replacing or repairing those components that have the greatest tendency of (1) failing prior to system failure and (2) contributing to system failure by being contained in a minimal cut set that causes

the system to fail. Preventive sensors can be used to detect these failures. By ranking of each component according to its sequential contributory importance (see Section 3.2.3.2), a designer can determine the components whose failures should be detected by sensors.

The scram control circuit for a TRIGA nuclear reactor given in Appendix C is redundant. There, the sequential contributory importance of each component is computed and plotted to show the optimal locations of preventive sensors in the circuit.

4.3.2 Diagnostic Sensors - We now consider systems in which there is a finite response time for operator action before a min cut set can cause system failure.

In this case, a fault tree can be an adequate model for describing the physical processes that result in an accident or system failure. The intermediate events can describe out-of-tolerance conditions that must occur if system failure is to occur. These events can be, however, detected in time by sensors. Thus, use of diagnostic sensors or monitors can arrest the propagation of failures.

For example, in Appendix D, a fault tree is given for a chemical processing system that describes a reactor explosion in terms of three subevents, (1) concentration of reactor stream too high, (2) temperature of reactor too high, and (3) reactor pressure too high. Any of these three events is sufficient to cause a reactor explosion. In Appendix D, we compute the importance of each of these events by the modular decomposition property to determine the subevent most critical to the occurrence of the top event. In this manner, we can determine the optimal location of diagnostic sensors in our system. In our example of Appendix

D, we have three choices regarding sensor location, (1) a flow meter for the reactant stream, (2) a temperature gauge for the reactor, and (3) a pressure gauge for the reactor. The example is an unpublished work by Yoon [85].

CHAPTER FIVE

FAULT TREES FOR DIAGNOSIS AND SIMULATION

Subsystem functional faults can produce catastrophic results if certain system conditions exist. For example, failure of an engineered safeguard system at a nuclear power plant can result in release of lethal radiation if a loss-of-coolant accident occurs. Another example, loss of a hydraulic system while a commercial jet is in flight, can result in loss of control of the aircraft. Fault tree analysis provides an efficient means of identifying subsystem functional faults. The information contained in the evaluation of the fault tree can assist an operator in making decisions that have a bearing on the safety and/or operability of the entire system when failure of a subsystem is observed.

In this chapter, we apply the concept of probabilistic importance to after-the-fact investigation. If a fault tree can accurately simulate system failure (i.e., if all failures can be described in terms of Boolean logic) then the fault tree can be quantitatively evaluated to determine the critical events. In the event of system/subsystem breakdown a repair checklist can be generated for an operator to follow. The basic events on the checklist can be ordered according to their importance when system failure occurs. In Sections 5.1 and 5.2 we present methods by which repair checklists can be generated. In Section 5.3 we present a checking scheme, based on the concept of criticality, that minimizes the expected time for system diagnosis. In Section 5.4 we discuss the choices available to an operator in the event system failure is observed and how decisions regarding system operation can be made based on a risk assessment. In Section 5.5 we describe how a fault tree can be utilized

as a simulation model for informational feedback during system fault conditions.

5.1 Generation of Repair Checklists

The appropriate measure of importance to use in generating repair checklists depends upon the type of system analyzed and its operating characteristics. We consider three types of systems separately, (1) passive standby systems such as emergency cooling systems, (2) continuously operating systems that are maintained; this includes most commercial operations such as power plants and chemical plants, and (3) operating systems that are not maintained during their mission life such as missile and satellite systems.

5.1.1 Standby Systems - Many safety systems are standby systems. They generally remain idle during their expected lifetime. There is a disturbing possibility that equipment, particularly passive components in these systems, can fail prior to demand and render the system inoperable. Critical standby systems such as engineered safeguard systems at a nuclear power plant, are tested periodically to decrease the likelihood that equipment will be unavailable upon demand (see Section 2.5.2.2.1) In this section, we show how to generate repair checklists in the event these systems fail to operate when tested. In Section 5.1.1.1, we show how to calculate the unavailability of components in standby systems. Finally, in Section 5.1.1.2, we consider the appropriate measure of importance to use in generating checklists for these systems.

5.1.1.1 Unavailability of Components in Standby Systems - If active components are tested frequently and maintained, it is reasonable to assume that their failure rate remains constant during the

system mission time. Active components in these systems must change state when called upon to operate, e.g., relay contacts must close, pumps must start, etc. Failure rate is described on a per-demand basis, i.e., by failures per cycle. The unavailability of these components is simply equal to their failure rate as shown in Section 2.8.2.2.1.

If testing the system at the end of some specified time interval (0, T), verifies that a passive component is working properly, then the probability of the component failing at T is simply F(T), where F is its life distribution. The unavailability of these components at test is simply F(T) where $F(T) = 1 - e^{-\lambda T} \approx \lambda T$ for $\lambda T \lesssim .01$. In this case, the failure rate is given on an hourly basis, e.g., failures/hr. In other instances the working state of a passive component may be verified at another inspection interval. If the length of this interval, T_I, is much smaller or much greater than the system inspection interval, T, then the component's unavailability can be calculated using the expression

$$\bar{A}_i = \lambda_i \, T_I/2 \qquad (5.1)$$

given in Section 2.5.2.2.1.

Table E-1 of Appendix E demonstrates how component unavailabilities for standby systems are calculated. In Appendix E, checklists are generated for the low pressure injection system (LPIS) which is a redundant standby safety system at a nuclear power plant. Technical specifications require that the LPIS be tested once a month. Each leg is tested by turning on a pump. Successful operation is verified by examining a pressure gauge. In Table E-1 the unavailability of all active components required to change state upon demand is simply given by their cyclic failure rates. The unavailability of passive components, such as wires

in the control circuit, pipe ruptures, etc., is given simply as λT where λ is their hourly failure rate and T their fault duration time, given as 720 hours (= one month). Also the unavailability of active components that are dormant (do not change state at test) but can disable the system through inadvertent actuation is also given as λT. For example, a normally-open motor-operated valve closing and preventing flow through a LPIS leg is such a component. In the LPIS, a pipe blockage or plugging can only be verified during refueling, which occurs once a year. The effective fault duration times for these events are given by 8760/2 where 8760 = number of hours in a year. Division by two results from relation (5.1). The AC and DC power systems required to operate the pump and open the valves are continuously operating maintained systems. Their unavailability is simply given by their steady state limiting unavailability.

5.1.1.2 Appropriate Measure of Importance for Standby System - It is clear from the discussion of the previous two sections that several cut sets can fail at test or on demand in standby systems. In this case, components contribute to, but do not necessarily cause, system failure. The assumption that a single component causes system failure in an instant of time is not valid because several dynamic components can fail to change state simultaneously. Hence, it is felt that the Vesely-Fussell definition of importance is suitable for ranking components in a standby system (see Section 3.2.3.4). Sequential measures of importance are not appropriate in this case.

5.1.2 Maintained Systems - For component failures that are statistically independent, it is a good assumption for a continously operating

system that system breakdown is caused by a component failing at some instant of time. The measures of importance that are suitable in ranking components in maintained systems are the steady-state sequential measures of importance, i.e., expression (3.16), the probability that a component causes system failure at steady state. The limiting expression for the sequential contributory importance can be obtained by a development similar to the one that led to equation (3.16):*

$$\frac{\sum_{j \atop i \neq j} [g(1_i, 1_j, \underline{A}) - g(1_i, 0_j, \underline{A})]\bar{A}_j/(\mu_j+\tau_j)}{\sum_{\ell=1}^{n} [g(1_\ell, \underline{A}) - g(0_\ell, \underline{A})]/(\mu_\ell+\tau_\ell)} \qquad (5.2)$$

In the chemical processing system analyzed in Appendix D, we used an expression similar to (3.16) in calculating the probability that a module in a fault tree causes system failure at steady state.

The sequential measures of importance give additional information regarding the failure history of a system, such as the most efficient way of diagnosing system failure. For example, a component contained in a cut set of order two may have a relatively high probability of causing the system to fail. In turn, the failure of this component may be difficult to check. The operator can have the option of checking the other components contained in the same min cut sets and determining indirectly whether this component has failed.

5.1.3 Non-maintained Systems - The same ideas apply to non-maintained systems when computing importance. The exception is that the

*Expressions (3.16) and (5.2) are time differential measures of importance rather than time integrated measures of importance.

sequential measures of importance are time dependent and are calculated in terms of the density, $f_i(t)$ (e.g., see expression (3.15)).

5.2 Checklist Generation Scheme

5.2.1 Practical Considerations - The occurrence of some basic events in a system may not be physically detectable. In the fault tree simulation of the system, the fault must propagate to a higher order event in the fault tree where its effect can be linked to some physically measurable quantities such as changes in temperature, pressure, flow rate, etc. to be detected. In this case, the fault tree must be modularized and higher order events (i.e., gate events) must be treated as basic events in the checklist.

In generating the checklist, false alarms should be considered, i.e., the reliability of the monitoring device that indicates system failure should be considered. In highly reliable systems, false alarms can be much more frequent than system failures so that the operator is "trained" to assume a false alarm.

5.2.2 Ordering of Basic Events on Checklist - The order in which the components are listed on the checklist should reflect the knowledge the operator gains about the system as he examines each component in the checklist. The ranking of the basic events should be done on a conditional basis. For example, if the operator finds that the first event has not occurred on the checklist, then the second event on the checklist should be the most critical to system failure, given that the first event has not occurred. In general, the i^{th} event is most critical to system failure given that the first $i - 1$ events have not occurred.

5.2.3 Sublist Generation - If a component, say i, in the checklist is found to be failed and is contained in a cut set of order two or higher, then a sublist is generated for component i. In the sublist we generate a ranking of cut sets containing component i by computing the probabilistic importance of these cut sets with component i failed. Again, we compute importance on a conditional basis. We then check the components in the cut sets that contain i. In general, it is unwise to include triple or higher order cut sets in the sublist. For maintained or inspected systems, the simultaneous occurrence of three independent events is rare. The author feels that the criteria adopted by the Reactor Safety Study are valid for checklist generation, i.e., retain the most important cut sets: (1) single passive faults, (2) single active faults, and (3) double active faults. If these criteria are adopted, the sublist is a single columnar list of active components ranked according to their probability of occurrence. By keeping only the most important cut sets, a multitude of trivial combinations that are normally given in a typical fault tree are eliminated from consideration. The purpose of the checklist is to aid the operator in making decisions that have to be made under a time constraint.

5.2.4 Dependent Events in a Checklist Generation - Though all basic events are assumed to be independent, dependent failures can be incorporated into the scheme by including basic events that cause secondary failures. On our checklist we can include basic events that describe environmental or operational conditions capable of simultaneously failing two or more system components. When we check for these secondary failure conditions, we generate a sublist for the components sensitive to these conditions.

5.2.5 Flowchart for Checklist Generation Scheme - The procedure that the operator must follow to examine the checklist is summarized in terms of a flow chart given in Figure 5.1. It shows that the checklist will change to reflect the increased knowledge concerning the system as time progresses.

5.2.6 Example of Checklist Generation Scheme - In Appendix E, we apply the checklist generation scheme of Figure 5.1 to a low-pressure injection system. As stated in Section 5.1.1.2, the appropriate measure used to rank basic events for standby systems is the Vesely-Fussell measure of importance.

5.3 System Diagnosis Under a Time Constraint

In Section 4.14, we considered upgrading systems under a cost constraint. The complementary problem in this chapter is system diagnosis under a time constraint. In Sections 5.1 and 5.2, we generated repair checklists solely on the basis of probabilistic importance. We did not consider the time required to check components. In some cases, there may be a considerable risk or system degradation while a system or subsystem is down. In this section, we propose a checking scheme that minimizes the expected time required to diagnose system failure based on the concept of component criticality. The scheme is based on an expression that is a function of the component checking times as well as their probabilistic importance. We now consider the restrictions and assumptions that apply to this expression as we derive it.

5.3.1 Expression to Minimize Checking Time - We assume that system failure is observed in some relatively small interval of time. It is

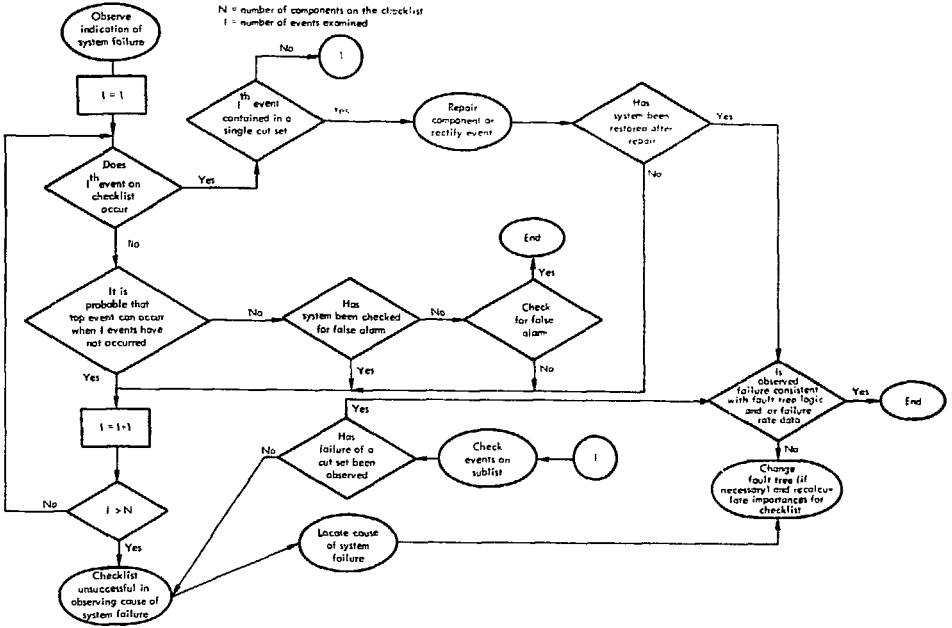


FIG. 5.1 Flow Chart for Checklist Generation Scheme

then reasonable to assume that if component failures are statistically independent, a cut set caused system failure and that one component is critical to system failure, i.e., $\Psi(1_i, \underline{Y}) - \Psi(0_i, \underline{Y}) = 1$. We check all components in the system one by one until failure of a critical cut set is observed (see Section 3.2.2.1).

First let us discuss a procedure in which a component is randomly chosen for checking. There are three possible outcomes regarding the state of the system as we check this component.

1. The component has not failed.
2. The component has failed but is not critical to system failure.
3. The component has failed and is critical to system failure.

If this component is chosen first to be checked and it is found to be failed, we stop checking only if the component is contained in a single-order cut set (i.e., it is in series with the rest of the system); otherwise we continue checking.

5.3.2 Notation - We adopt the notation of Section 3.8. In addition, let T_i denote the time required to check component i; $q_i(t) \equiv q_i$; $p_i = 1 - q_i$, T_s = time to diagnose system failure; $(1^K, 0^{n-K}, \underline{Y}^{N-n})$ be the state vector of a system comprised of N component where n components have been checked, $n \leq N$, K component have been found to be failed and n-K components are not failed; let $C'(\underline{Y})$ denote the set of components that have been checked and $C°(\underline{Y})$ the set of components that have not been checked.

5.3.3 Derivation - An expression for the expected time to diagnose system failure, $E[T_s]$, involves $\sum_{i=1}^{n} 2^{i-1}$ terms where N = number of components and i is the order. The first seven terms according to order are given by

$$E[T_s] = T_1 + \underbrace{\begin{Bmatrix} T_2\, p_1\, \Delta_2 g(0_1, \underline{q}) \\ T_2\, q_1\, \Delta_2 g(1_1, \underline{q}) \end{Bmatrix}}_{\text{Second Order}} + \underbrace{\begin{bmatrix} T_3\, p_1\, p_2 \Delta_3 g(0_1, 0_2, \underline{q}) \\ T_3\, p_1\, q_2 \Delta_3 g(0_1, 1_2, \underline{q}) \\ T_3\, q_1\, p_2 \Delta_3 g(1_1, 0_2, \underline{q}) \\ T_3\, q_1\, q_2 \Delta_3 g(1_1, 1_2, \underline{q}) \end{bmatrix}}_{\text{Third Order}} + \ldots$$

$\underbrace{}_{\text{First Order}}$

where the terms following the vertical brackets are summed.

There are 2^n possible arrangements involving $E[T_s]$. Note that in the ordering given above, if we check component 2 first and component 1 second, the terms involving $T_3, \ldots T_n$ do not change. To determine which component to check first, we minimize $E[T_s]$ with respect to the first two terms and neglect third and higher order terms since they have no effect in finding the minimum in this case. If

$$T_2 + \begin{Bmatrix} T_1 p_2 \Delta_1 g(0_2, \underline{q}) \\ T_1 q_2 \Delta_1 g(1_2, \underline{q}) \end{Bmatrix} > T_1 + \begin{Bmatrix} T_2 p_1 \Delta_2 g(0_1, \underline{q}) \\ T_2 p_1 \Delta_2 g(1_1, \underline{q}) \end{Bmatrix} \tag{5.3}$$

then component 1 should be checked before component 2 and, in general, if

$$T_j + \begin{Bmatrix} T_j p_j \Delta_i g(0_j, \underline{q}) \\ T_i q_j \Delta_i g(1_j, \underline{q}) \end{Bmatrix} > T_i + \begin{Bmatrix} T_j p_i \Delta_j g(0_i, \underline{q}) \\ T_j q_i \Delta_j g(1_i, \underline{q}) \end{Bmatrix} \tag{5.4}$$

for all $j(\neq i)$, then component i should be checked first. The argument can be extended each time we check a component in the system. In general, if we have checked n components in the system, the next component we

should check is again determined by an expression similar to (5.4)

$$T_j + \begin{cases} T_i p_j \Delta_i g(0_j, 1^k, 0^{n-k}, \underline{y}^{N-n}) \\ T_i q_j \Delta_i g(1_j, 1^k, 0^{n-k}, \underline{y}^{N-n}) \end{cases} > T_i + \begin{cases} T_j p_i \Delta_j g(0_i, 1^k, 0^{n-k}, \underline{y}^{N-n}) \\ T_j q_i \Delta_j g(1_i, 1^k, 0^{n-k}, \underline{y}^{N-n}) \end{cases} \tag{5.5}$$

where $i \,\&\, j \in C^o(\underline{Y})$. The optimization procedure in expression (5.5) is referred to in decision theory as a one-step-ahead optimization policy. [62]

5.3.4 Series System - Let us use expression (5.4) to determine which component should be checked first for a series system with N components.

In this case,

$$g(\underline{q}) = 1 - \prod_{i=1}^{N} (1 - q_i) = 1 - \prod_{i=1}^{N} p_i,$$

then (5.4) becomes

$$T_j + \begin{cases} T_i p_j \prod\limits_{\substack{k \\ k \neq i \\ k \neq j}} p_k \\ T_i q_j \cdot 0 \end{cases} > T_j + \begin{cases} T_j p_i \prod\limits_{\substack{k \\ k \neq i \\ k \neq j}} p_k \\ T_j q_i \cdot 0. \end{cases}$$

This implies that

$$T_j + T_i p_j \prod_{\substack{k \\ k \neq i \\ k \neq j}} p_k > T_i + T_j p_i \prod_{\substack{k \\ k \neq i \\ k \neq j}} p_k$$

for reliable systems $\prod\limits_{\substack{k \\ k \neq i \\ k \neq j}} p_k \approx 1$

$$T_j + T_i p_j > T_i + T_j p_i$$

or

$$\frac{T_j}{q_j} > \frac{T_i}{q_i} \quad \text{for all } j \neq i.$$

The above inequality states that the component with the minimum value of $\frac{T_i}{q_i}$ should be checked first, an intuitive result for a series system.

5.3.5 Parallel System - For a parallel system, $g(\underline{q}) = \prod_{\ell=1}^{N} q_\ell$, (5.4) becomes

$$T_j + T_i q_j \prod_{\substack{\ell \neq i \\ \ell \neq j}} q_\ell > T_i + T_j q_i \prod_{\substack{\ell \neq i \\ \ell \neq j}} q_\ell$$

$$T_j\left(1 - \prod_{\ell \neq j} q_\ell\right) > T_i\left(1 - \prod_{\ell \neq i} q_\ell\right)$$

for reliable systems $\prod_{\ell \neq i} q_\ell \approx 0$, $\prod_{\ell \neq j} q_\ell \approx 0$, which implies that

$$T_j > T_i \quad \text{for all } j \neq i.$$

For a parallel system, the above inequality says that the component with the minimum check time should be checked first, again an intuitive result.

A disadvantage to the above scheme is that it maximizes $E[T_s]$ with respect to the first two terms only. Third and higher order terms may have to be considered in finding the true optimal checking order. The author conjectures that it is extremely difficult to set up a generalized expression that minimizes $E[T_s]$. Expression (5.5) is easy to compute and gives intuitive results for the series and parallel cases.

5.4 Decisions Regarding System Operation Based on Risk Assessments

After the operator has identified the basic events such as hardware failures and maintenance faults that have occurred, the increased risk of operating or the system degradation can be determined by quantitatively evaluating the fault tree for the entire system. On the basis of such factors as (1) the length of time it may take to repair components or rectify human errors or (2) the severity associated with loss of subsystem or component, decisions may be made regarding the operation while repairing components (3) operate system and simultaneously repair or (4) operate the system without repair. For example, all four choices are, in principle, available to an operator at a nuclear power plant if an engineered safeguard system is found inoperable. Choices (1) and (4) are available to a pilot who finds a hydraulic system inoperable in flight, i.e., he may land his aircraft at the nearest airport or continue his flight to his final destination.

5.4.1 Shutdown Decision at a Nuclear Power Plant - As an example of a decision to be made on a risk-assessment basis, consider a failure of low-pressure injection pump A revealed during its monthly test (see Appendix E). The operator would like to know if this failure warrants plant shutdown. Technical specifications require the plant to be shutdown to a hot standby condition if repair takes longer than 24 hours, i.e., $T > 24$ hours, and to a cold standby condition if $T > 48$ hours. The effect of the failure of pump A means that leg A is incapacitated until pump A can be fixed. That means that the LPIS system has lost its redundancy. If a double ended pipe rupture should occur and the leg B pump should fail to start, the potential exists for a large radiological release.

There is, however, also a finite risk associated with plant shutdown. In the next section we use the quantitative information presented in the Reactor Safety Study and in Appendix E to compute the risk of shutting down the plant versus the risk of plant operation with one LPIS pump out of service. We include the effect of thermal transients induced by shutdown and startup. We then determine the time interval T for which the risk associated with plant operation becomes comparable to the risk of shutting the plant down. By such a determination the maximum allowable repair time T can be established.

5.4.1.1 Establishing Maximum Allowable Repair Time, T - From Appendix E, Table E-2, we see that with one LPIS pump out of service, the probability that the entire LPIS fails on demand is 7.949×10^{-3}. From Table 2-4, the probability of a large pipe break is 10^{-4}/yr.* The hourly risk then associated with plant operation with one LPIS pump out of service is

$$\text{Prob (radiological release/hr} \mid \text{one LPIS pump failure)} = \lambda \text{ (large pipe break/hr)} * \text{Prob (LPIS failure} \mid \text{one LPIS pump failure)}$$

$$= 10^{-4}\text{/yr} \times (1 \text{ yr}/8760 \text{ hrs}) \times 7.949 \times 10^{-3}$$

$$= 9.0742 \times 10^{-11}\text{/hr.} \qquad (5.6)$$

In the case of a PWR, the Reactor Safety Study considered accident chains with loss of offsite power as an initiating event. They considered that this accident sequence significantly contributed to the overall risk of

*In the event of a small pipe break, the high pressure injection system can provide emergency cooling.

nuclear power plant operation. If both the main feedwater and auxiliary feedwater systems fail to operate following this transient, the heat sink is lost for decay heat removal. The steam generators would be emptied in about 1/2 hour, causing the reactor coolant in the primary loop to heat up. The reactor coolant would be discharged through the pressurizer relief valves causing the reactor core to be uncovered. Within approximately 1-1/2 hours after the transient, core melting would start. Various accident sequences were hypothesized that would result in loss of the main feed water and auxiliary feedwater systems with loss of offsite power as the initiating event. As shown in Table 5-1, these sequences make a significant probability contribution across the entire release spectrum.

TABLE 5-1

Transient Event Probability Contribution

Release Category R	$P_{TE,R}$	$P_{TE,R}/P_{TOTAL,\ R} \times 100\%$
1	9×10^{-8}	1%
2	5×10^{-7}	1%
3	2×10^{-7}	.4%
4	6×10^{-8}	1 %
5	4×10^{-7}	3%
6	4×10^{-6}	10%
7	8×10^{-6}	2.66%

where $P_{TE,R}$ = probability per year that an accident sequence with the initiating event "loss of offsite power" results in the loss of the heat removal systems, a core melt and the indicated release.

$P_{TOTAL,R}$ = probability per year that the indicated release occurs from all causes with initiating events, large LOCA, small LOCA, reactor vessel rupture, transients events, etc.

The probability of the accident sequence, P_{TE}, took the general form

$$P_{TE} = P_1 \prod_{i=2}^{n} P_i \tag{5.7}$$

where P_1 = probability that loss of offsite power occurs during normal operation = .2 occurrences/year

P_i = probability that the i^{th} event in the accident sequence occurs.

For our example, we are concerned that during the scheduled shutdown of the plant, an operator error is committed that causes a turbine trip, which in turn imposes a transient instability in the electrical grid network resulting in loss of offsite power. We estimate that the probability of operator error during shutdown causing a turbine trip is 10^{-2}. Based on Federal Power Commission data, the probability that offsite power is lost during a turbine trip is 10^{-3}. [72] The probability that an operator error is committed during shutdown causing loss of offsite power is obtained by multiplying $P_{TE,R}$ in Table 5-1 by the ratio

$$\frac{10^{-2} \times 10^{-3}}{.2} = 5 \times 10^{-5}.$$

The probability of a radiological release caused by an operator error described above is

$$5 \times 10^{-5} \sum_{R=1}^{7} P_{TE,R} = 7.0 \times 10^{-10}. \tag{5.8}$$

During shutdown and startup thermal transients may increase the probability of the loss of coolant accident (LOCA). We assume that the probability of the LOCA is the same during shutdown and startup but different from the value during steady state generation. We define α as

$$\alpha = \frac{\lambda \text{ (LOCA during SHUTDOWN OR STARTUP)}}{\lambda \text{ (LOCA at STEADY STATE)}}.$$

After repair or replacement of the LPIS pump, plant startup begins. The LPIS unavailability is given in Table E-2 as 5.064×10^{-4}. We estimate the time required to shut the plant down or start it up as 24 hours. The increased risk due to shutdown and startup is

$$(\alpha-1)(10^{-4}/\text{yr})(1 \text{ yr}/8760 \text{ hrs}) [7.949 \times 10^{-3} + 5.064 \times 10^{-4}] \text{ 24 hrs.}$$

$$= (\alpha-1)\, 2.32 \times 10^{-9}. \qquad (5.9)$$

We eliminate failure of other engineered safeguard systems to simplify the analysis.* From (5.6), (5.8) and (5.9), we express the allowed repair time, T, as a function of α, by

$$9.0742 \times 10^{-11}\, T = 7.0 \times 10^{-10} + (\alpha-1)\, 2.32 \times 10^{-9}. \quad (5.10)$$

Equation (5.10) is plotted in Figure 5.1. We see that if $\alpha = 1$, T is given by 7.7 hours, a lower bound in this analysis. The actual value of α will dictate the value for T. The above analysis not intended to be rigorous. It does show, however, how decisions regarding system operation based on a risk assessment can be made when system fault conditions occur.

*The unavailability of the LPIS with one pump out of service should dominate all other engineered safeguard system unavailabilities, making expression (5.9) an accurate approximation.

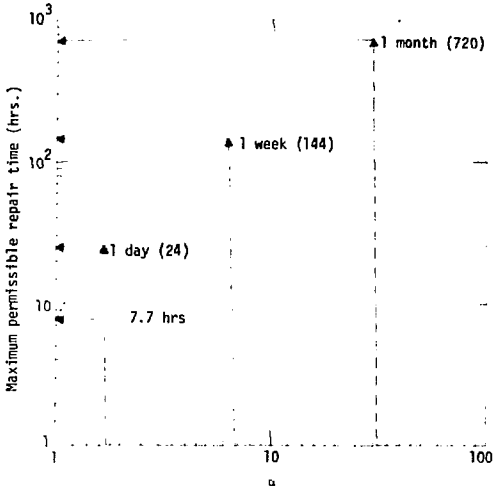


FIG. 5.2 Determination of Maximum Permissible Repair Time

5.5 Utilization of Fault Tree Simulation for Informational Feedback During System Fault Conditions

We combine all the concepts presented in this chapter to show how fault tree analysis can be applied in the operational phase of a system. In this section, we consider fully maintained systems at steady state. We devise an algorithm to show how fault tree logic can be programmed in a computer and through teletype communication to assist an operator in making decisions and initiating actions that have bearing on safety.

In Appendix D, we described how a fault tree of a chemical processing system could be modularized into subevents. These subevents described out-of-tolerance conditions whose occurrence can be detected by a sensing device. This implies that the effect of these conditions can be measured and monitored on an instrument panel. This further implies that a fault tree, in some cases, can be used as a simulation model in forewarning the operator of potentially catastrophic fault conditions. We show in this section how the system can be efficiently diagnosed to determine the cause of system failure when these conditions occur. Then decisions regarding system operation based on risk assessments can be made as described in Section 5.4.

We now identify two types of fault events in fault tree simulation.

5.5.1 Fault Events in Fault Tree Simulation - One type of fault event to be considered is an event that must be combined with at least one other primary event in the fault tree if the top event is to occur. This implies that this fault event is an input to an AND gate at a higher level in the fault tree. We call these fault events, properly contained fault events, because the min cut sets to these events are properly contained in min cut sets for the top event. For these fault events, we show how probabilistic importance can be computed to identify components whose failures are critical to system failure. In this manner, we can reveal the necessary components which must not fail if system failure is to be prevented and the accident avoided.

We also consider a second type of fault event that can, by itself, cause system failure, i.e., there is all OR logic associated with propagating the fault event to the top event. We call these fault events

self-propagating fault events. If the response time of these fault events are greater than the time required for operator action in averting system failure, then the top undesired event can be avoided. For some self-propagating fault events, there may be an adequate amount of time to examine the system to determine the components that have failed before deciding on the mode of operation while the system is being repaired. In Section 5.5.1.2, we show how the expected checking time to diagnose system failure can be determined for self-propagating fault events. If the response time to the cut sets of these fault events is known, we can establish whether there is a sufficient amount of time for checking before deciding on the proper course of operator action.

5.5.1.1 Properly Contained Fault Events - From the previous discussion, fault events that cannot propagate by themselves to the top event are called properly contained fault events. When these fault events occur, the following information can be provided in assisting the operator in making decisions regarding the future operation of the system: (1) the basic events most critical to system failure when the fault event occurs and (2) the mean time to system failure when the fault event occurs.

5.5.1.1.1 Importance Ranking to Determine Critical Components - For continuously operating systems, we stated in Section 5.1.2 that the appropriate measures of importance to rank basic events are the sequential measures of importance. In Appendix C, we mentioned how the sequential contributory importance measure can be used to locate sensors in a system. We claimed that, for redundant systems, the components that have the greatest tendency of failing prior to system failure should be monitored.

In this section, we consider the opposite situation. Given that some intermediate fault events, M, has occurred, what are the basic events expected to occur if system failure is to occur? For a maintained system at steady state, we can determine these critical basic events by setting the indicator variable of the fault event equal to one, $Y^M = 1$, and then rank basic events by the steady-state Barlow-Proschan measure of importance, given below (see expression 3.16)

$$\frac{[g(1_i, 1^M, \underline{A}) - g(0_i, 1^M, \underline{A})]/(\mu_i+\tau_i)}{\sum_{j=1}^{n}[g(1_j, 1^M, \underline{A}) - g(0_j, 1^M, \underline{A})]/(\mu_j+\tau_j)} . \tag{5.11}$$

Using expression (5.11), we can monitor the critical components while system diagnosis and repair is taking place.

5.5.1.1.2 Mean Time to System Failure - The mean time to system failure when fault event M occurs is given by an expression similar to expression (2.44)

$$MTFF > \frac{1}{\sum_{i=1}^{n}\frac{\ln[1 - \Delta g_i(1^M, \underline{A})]}{(\mu_i+\tau_i)}} \tag{5.12}$$

where $\Delta g_i(1^M, \underline{A}) = g(1_i, 1^M, \underline{A}) - g(0_i, 1^M, \underline{A})$.

Expression (5.12) is an indication of the amount of time available to an operator for system diagnosis when a non-propagating fault event occurs.

5.5.1.2 Self-Propagating Fault Events - We now consider fault events whose min cut sets are min cut sets for the top event, i.e., self-propagating fault events.

If we know the response time of all min cut sets for these fault events, and the checking time required for all basic events in these cut sets, then we can determine whether there is enough time for operator action. There are basically two distinct choices regarding operator action when system fault conditions occur, (1) immediate remedial action and (2) system diagnosis followed by remedial action.* The choice depends obviously on the expected response time of the fault event in causing the top event to occur. In the following section, we show what action should be taken if a self-propagating fault event occurs.

5.5.1.2.1 Response Time Probabilities for Self-Propagating Events - In this section, we derive the following two expressions, (1) the probability that there is sufficient time for an operator to take immediate remedial action and (2) the probability that there is sufficient time to diagnose the cause of system failure. The determination of these probabilities will tell the operator the choices available to him when a self-propagating fault event occurs.

We now present the notation used to derive these probability expressions:

Notation: Let M denote a self-propagating fault event; K_j be a minimal cut set contained in M; T_M^{IRA} be the time required for immediate remedial action when M occurs; $T_{K_j}^{D}$ be the checking time required to verify that min cut set, K_j has occurred; let $T_{K_j}^{RS}$ be the response time for cut set K_j to cause system failure. Let Y^M be the indicator variable for M with $E[Y^M] = h(\underline{A})$.

*Immediate remedial action is any action that can be accomplished in a relatively short period of time; examples include (1) pushing a scram button, (2) closing a valve and (3) closing a circuit breaker.

5.5.1.2.1.1 Derivation of Immediate Remedial Action Probability - The probability that a basic event i, $i \epsilon M$, causes M to occur in the steady state, given that M just occurred is given

$$\frac{[h(1_i, \underline{\bar{A}}) - h(0_i, \underline{\bar{A}})]/(\mu_i+\tau_i)}{\sum_{j \epsilon M}[h(1_j, \underline{\bar{A}}) - h(0_j, \underline{\bar{A}})]/(\mu_j+\tau_j)}. \tag{5.13}$$

If the rare event approximation is valid, then (5.13) becomes

$$\frac{\left[\sum_{j} \prod_{\substack{i \epsilon K_j \\ \ell \epsilon K_j^i \\ Y_i=1}} \bar{A}_\ell\right] / (\mu_i+\tau_i)}{\sum_{j \epsilon M}[h(1_j, \underline{\bar{A}}) - h(0_j, \underline{\bar{A}})]/(\mu_j+\tau_j)} \tag{5.14}$$

where $K_j \epsilon M$. Expression (5.14) follows from a derivation given in Section 4.1.5. Let

$$Y_{K_j}^{IRA} = \begin{cases} 1 \text{ if } T_M^{IRA} \geq T_{K_j}^{RS} \\ 0 \text{ if } T_M^{IRA} < T_{K_j}^{RS} \end{cases}.$$

When a self-propagating event occurs, the probability that the operator cannot take immediate remedial action is given by

$$\frac{\sum_{i \epsilon M}\left[\sum_{j} \prod_{\substack{i \epsilon K_j \\ \ell \epsilon K_j^i \\ Y_i=1}} \bar{A}_\ell \, Y_{K_j}^{IRA}\right] / (\mu_i+\tau_i)}{\sum_{j \epsilon M}[h(1_j, \underline{\bar{A}}) - h(0_j, \underline{\bar{A}})]/(\mu_j+\tau_j)}. \tag{5.15}$$

5.5.1.2.1.2 Derivation of System Diagnosis Probability - The probability that an operator has sufficient time to diagnose failure, i.e., find out what cut set has failed, is a more difficult determination. We assume the operator can interact with a computer. Furthermore, we assume that a computer program is set up that determines an optimal checking scheme that minimizes the time required to diagnose system failure as described in section 5.3.

The order in which components are checked is determined by expression (5.5). For each cut set K_j we set up the vector $(\underline{0}^{i\notin K_j}, \underline{1}^{i\in K_j})$. We use expression (5.5) successively, until on the n^{th} step, we observed that min cut set K_j has occurred, i.e., $(\underline{0}^{n-|K_j|}, \underline{1}^{|K_j|}, Y^{N-n})$ where $|K_j|$ is the number of basic events in K_j. The expected time to diagnose system failure when K_j occurs is given by

$$E[T_K^D] = \sum_{i\in C_{K_j}^1(\underline{Y})} T_i$$

where $C_{K_j}^1(\underline{Y})$ is the set of components which must be checked to determine K_j has caused system failure and T_i is the check time required for basic event i. Let

$$Y_{K_j}^D = \begin{cases} 1 \text{ if } T_{K_j}^D \geq T_{K_j}^{RS} \\ 0 \text{ if } T_{K_j}^D < T_{K_j}^{RS} \end{cases}$$

then the probability that the operator does not have sufficient time to diagnose system failure when M occurs is given by

$$\frac{\sum_{i\in M}\left[\sum_{j}\prod_{\substack{i\in K_j\\ \ell\in K_j\\ Y_j=1}}\bar{A}_\ell Y^D_{K_j}\right]/(\mu_i+\tau_i)}{\sum_{j\in M}[h(1_j,\underline{\bar{A}})-h(0_i,\underline{\bar{A}})]/(\mu_j+\tau_j)}. \tag{5.16}$$

A predetermined course of action can be prescribed by expressions (5.15) and (5.16) when self-propagating fault events occur. Expression (5.16) indicates if there is enough time to diagnose the cause of system failure; if there is not, then we determine by (5.15) if there is an adequate amount of time for immediate remedial action. If the fault event propagates instantaneously, then an automatic system response is required to avert system failure.

5.5.2 The Occurrence of Two or More Cut Sets - We assumed in Section 5.1.2, that for a continuously-operating maintained system, system breakdown is caused by a component failing at some instant of time. The possibility exists that two or more cut sets can occur when a component causes the system to fail.

For example, let us assume that it is observed that a cut set, say K_j, of order two or higher caused the system to fail. If we can establish which component, say i, actually caused the system to fail, then we can generate a listing of other cut sets containing i that can also occur. We can do so on the basis of the following expression,

$$\frac{[g(\underline{1}^{K_j},\underline{1}^{K_\ell},\underline{\bar{A}})-g(0_i,\underline{1}^{K_j-\{i\}},\underline{1}^{K_\ell-\{i\}},\underline{\bar{A}})]\prod_{\substack{m\in K_\ell\\ m\neq i}}\bar{A}_m/(\mu_i+\tau_i)}{[g(\underline{1}^{K_j},\underline{\bar{A}})-g(0_i,\underline{1}^{K_j-\{i\}},\underline{\bar{A}})]\prod_{\substack{n\in K_j\\ n\neq i}}\bar{A}_n/(\mu_i+\tau_i)} \tag{5.17}$$

where $i \epsilon K_j$ and $i \epsilon K_\ell$. We are conditioning on the event that i caused system failure with K_j, one of the cut sets that caused system failure.

5.5.3 Flowchart for Computer-Operator Interaction - We now describe an algorithm presented in Fig. 5.3 which is suitable for computer implementation. The algorithm shows how the operator can interact with a computer in diagnosing system failure with fault tree logic. The expressions presented in Sections 5.5 and beyond are evaluated in the computer as the operator provides teletype input.

Description of Algorithm

The computer stores in memory the cut sets and failure rate data. When a fault condition occurs, the operator inputs all known parameters into the computer. The computer identifies that a fault event occurs or asks for additional information. The computer identifies a fault event as either a self-propagating fault event or a properly-contained fault event.

In the case of a properly-contained fault event, the computer prints out the vital data as described in Section 5.5.1.1, i.e., (1) the mean time to system failure and (2) a listing of critical components that require monitoring.

If a self-propagating fault event occurs, then the computer tells the operator if there is adequate time for checking. If there is not, the computer tells the operator about the immediate remedial action required.

If there is a sufficient amount of time for checking, the computer asks for any known component failures. On the basis of this information, the computer lists the most important events that should be checked first.

If the time required to check is limited, then expression (5.5) is used to generate the checklist. If the most important events have not occurred, then the computer lists the second most important events by the iteration process described in Appendix E. This iteration process uses the information the operator gains as he examines the system. The operator interacts with the computer via teletype communication to inform the computer of all components that have been found to be failed during the checking process. The computer continues the iteration process until the occurrence of a min cut set has been observed or a false alarm has been diagnosed.

If the operator observes some environmental condition that has occurred, then he checks all components sensitive to this environmental condition. He also checks for any other min cut sets that may have occurred after establishing the cause of system failure. Based on a risk assessment as described in Section 5.4, the computer informs the operator of the proper course of system operation.

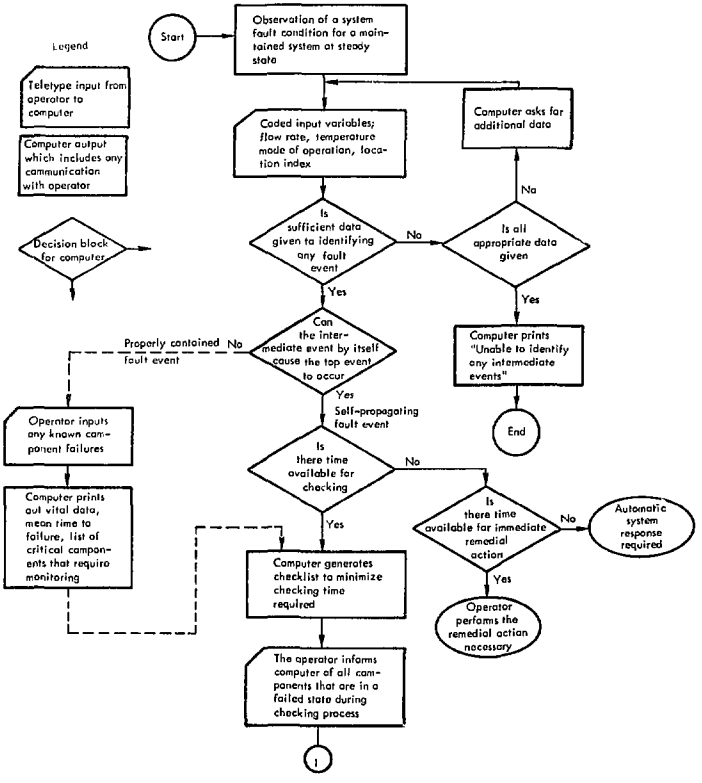


FIG. 5.3 Flowchart for Operator-Computer Interaction Simulation Process

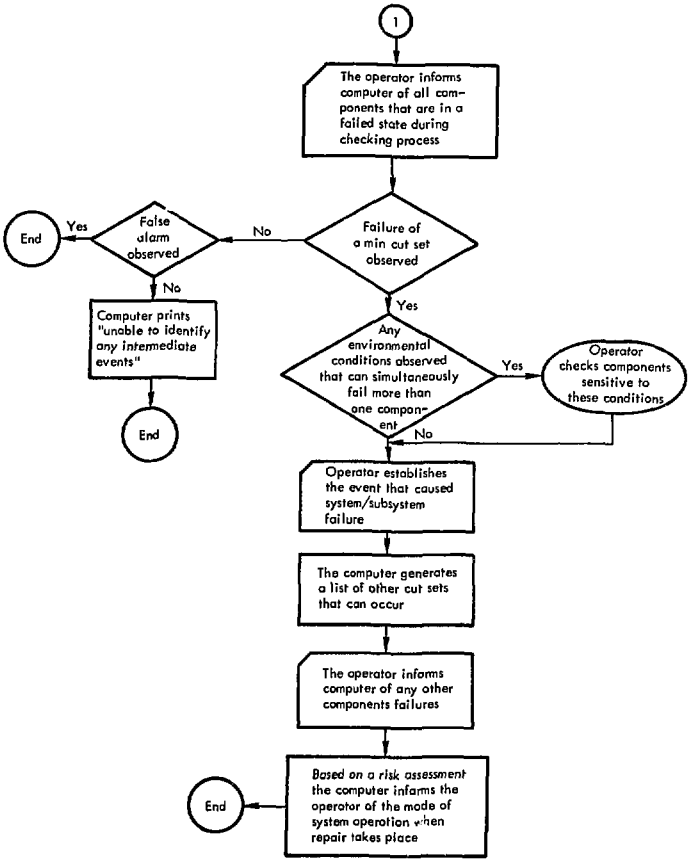

1
The operator informs computer of all components that are in a failed state during checking process
Failure of a min cut set observed
No
False alarm observed
Yes
End
No
Computer prints "unable to identify any intermediate events"
End
Yes
Any environmental conditions observed that can simultaneously fail more than one component
Yes
Operator checks components sensitive to these conditions
No
Operator establishes the event that caused system/subsystem failure
The computer generates a list of other cut sets that can occur
The operator informs computer of any other components failures
Based on a risk assessment the computer informs the operator of the mode of system operation when repair takes place
End

FIG. 5.3 (Cont'd.)

CHAPTER SIX

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

6.1 Conclusions

6.1.1 Application of System Safety Techniques - The first method to be applied in any safety program is a preliminary hazards analysis. The prime objectives of a PHA are to identify, minimize and control hazards. A PHA is ideally performed at the conceptual stages of the system life cycle, though in practice it can be applied later and still accomplish its objectives. A failure-mode-and-effects analysis should be initially performed at the design stage of the system. At that point, a FMEA can identify any single hardware failure modes that are critical with respect to the system's safety and/or reliability. A criticality analysis can rank these failure modes according to their probability of being critical with respect to system failure. Component criticality can be reduced by design changes at either the system or component level. Fault tree analysis is best applied during the detailed design stages of a system. FTA is particularly efficient in identifying basic causes such as hardware failures, human error, and environmental conditions that can cause subsystem functional faults to occur. The structuring of the fault tree at the top level provides an efficient format for describing the accident phenomenology associated with the top undesired event. An alternate representation for top level fault trees are event trees.

6.1.2 FTA versus FMEA - FMEA is a much simpler technique to apply than FTA. FMEA in many cases is the most cost effective technique to apply in analyzing small systems when a single failure analysis is

adequate. FTA in many cases is difficult to apply, is costly and time consuming. Its results are difficult to check. However, as systems become more complex and the consequences of accidents become catastrophic, a technique such as FTA should be applied. Inductive analysis can become extremely inefficient when analyzing complex systems due to the large number of component states that must be considered. FTA can efficiently direct the efforts of an analyst in considering only those basic events that can contribute to system failure, i.e., to the occurrence of the top event. FTA can efficiently represent the relationship of human error and environmental conditions in causing system failure. Actually the information in a FMEA is required at the component level in the fault tree. The two techniques FMEA and FTA complement each other.

6.1.3 Disadvantages to FTA - A major disadvantage to FTA is the possibility of oversight and omission. Automated fault tree construction can eliminate the possibility of omitting the routine failure modes. The automated approach can standardize fault tree analysis and eliminate the confusion associated with the seemingly different ways analysts can manually construct fault trees.

A problem in fault tree modeling is that it is difficult to apply Boolean logic to describe failures of system components that can be partially successful in operation and thereby having effects on the performance of the system.

Leakage through a heat exchanger is a good example. In addressing the partial failure problem, an analyst may have to describe the process analyzed in terms of the basic laws of mass, energy and heat balances as chemical engineers do in process simulation.

6.1.4 Probabilistic Importance and Applications - A fundamental quantity in computing probabilistic importance is Birnbaum's measure of importance $g(1_i, \underline{q}) - g(0_i, \underline{q})$, the probability that the system is in a state in which the occurrence of event i is critical to system failure. Two measures of importance were described; (1) measures that depend upon one point in time and are not a function of past behavior of the system, and (2) measures that are functions of the sequences of events that cause system failure. Measures of the second type give additional insight into system behavior not available with measures of the first type. The appropriate measure of importance to use in reliability engineering applications depends upon the time system failure is observed and on the type of system analyzed. In this thesis, importance was applied to areas of system design and diagnosis. The specific applications included:

1) Upgrading systems designs
2) Location of preventive and diagnostic sensors in a system
3) Generation of repair checklists
4) Simulation of System Failure by fault tree logic.

6.1.5 Quantitative Fault Tree Analysis - If basic events are statistically independent, then the min cut upper bound is an accurate approximation for the probability of the top event. For maintained systems, the expected number of system failures for small time is an accurate approximation for $F_s(t)$, the distribution of time to first failure. For large time, it appears that the T* method is an accuate approximation of $F_s(t)$ at least in the case of constant failure and repair rates. The steady-state upper bound provides a simple and direct means of computing the mean

time to first failure for a maintained system.

A major difficulty with quantitative fault tree evaluation is the lack of pertinent failure rate data. Even in cases where the data are good, it is not clear that we can justifiably apply to one system environment data that were obtained in a different system environment. In addition, the analyst might inadvertently apply inapplicable failure rate data; e.g., an hourly failure rate to a cyclic event. The human element is in itself difficult to quantify.

Nevertheless, quantitative evaluations are particularly valuable for comparing systems designs that have similar components. The results are not as sensitive to the failure rate data as is an absolute determination of the system failure probability. Because of uncertainties in failure rate data, quantitative fault analysis has its greatest value when relative rather than absolute determinations are made. As an initial estimate of the failure rate, proportional hazards can be assumed, i.e., the assumption that the failure rate, $\lambda(t)$, has the same time dependent behavior for all basic events. In the case of maintained systems, relative determinations can be made if all failure rates and repair rates are expressed in terms of a reference time unit.

Relative determinations can make qualitative judgments quantitative. The analyst by inspecting the minimal cut sets can rank basic events according to their relative frequency of occurrence. For example, an analyst may estimate that failure of a motor-operated valve to open upon demand is 1000 times more likely to occur than the rupture of that same valve. Such estimates can carry qualitative decision making one step further by permitting the importance of the most critical basic events to be plotted. These plots provide a more powerful form of decision making

In the context of qualitative versus quantitative decision making, the author fully concurs with a statement made by Lord Kelvin [55]

> *"I often say ... that when you can measure what you are speaking about, and express it in numbers, you know something about it; but when you cannot measure it, when you cannot express it in numbers, your knowledge is of a meagre and unsatisfactory kind; it may be the beginning of knowledge, but you have scarcely, in your thoughts, advanced to the stages of science, whatever the matter may be."*

6.2 Recommendations for Future Work

Since the steady state upper bound is presented in this thesis without proof, it would be useful to show the classes of failure and repair distributions for which the steady-state upper bound is applicable.

It would also be useful to incorporate an option in the IMPORTANCE computer code to allow for an error analysis. This can be accomplished by placing prior distributions on the failure rate data and then use Monte Carlo simulation to determine the spread in the importance rankings.

A program to generate repair checklists from fault trees can be written by a simple extension of the programming methods and algorithms given in the IMPORTANCE code. A more difficult task would be to program the fault tree simulation model given in Chapter Five. An interesting research problem would be to establish the feasibility and usefulness of such a program in diagnosing failure in real world systems.

REFERENCES

[1] M. Acero, Fault Tree Analysis of Reactor Safety Systems, MS Thesis, Dept. of Nucl. Engr., Univ. of Calif., Berkeley (1974).

[2] S. N. Anykora, et al, "Some Data on the Reliability of Instruments in the Chemical Plant Environment," The Chemical Engineer, (1971).

[3] G. Apostolakis, Mathematical Methods of Probabilistic Safety Analysis, School of Engineering and Applied Science, UCLA, Rept. UCLA-ENG-7464 (1974).

[4] R. E. Barlow and F. Proschan, Importance of System Components and Fault Tree Analysis, Operations Research Center, Univ. of Calif., Berkeley, Rept. ORC 74-3 (1974).

[5] R. E. Barlow and F. Proschan, Mathematical Theory of Reliability, (John Wiley and Sons, New York for SIAM, 1965) p. 146.

[6] R. E. Barlow and F. Proschan, Statistical Theory of Reliability and Life Testing, (Holt, Rinehart, and Winston, New York, 1975)

[7] R. E. Barlow and F. Proschan, Theory of Maintained Systems: Distribution of Time to First Failure, Operations Research Center, Univ. of Calif., Berkeley (1974).

[8] Z. W. Birnbaum, "On the Importance of Different Components and a Multicomponent System," Multivariate Analysis-II, P.R. Krishnaiah, Editor, Academic Press, New York (1969).

[9] M. Brown, "The First Passage Time Distribution for a Parallel Exponential System with Repair," (in Reliability and Fault Tree Analysis, R. E. Barlow, H. B. Fussell and N. D. Singpurwalla, editors, SIAM, 1975).

[10] P. Chatterjee, Fault Tree Analysis: Reliability Theory and Systems Safety Analysis, Operations Research Center, Univ, of Calif., Berkeley, Rept, ORC 74-34 (1974).

[11] K. E. Coulter, et al., "Improving Onstream Time in Process Plants," Chem. Engr. Progr. 68 (3), 56-59 (1972).

[12] D. R. Cox, Renewal Theory, (Methuen, London, 1962).

[13] G. E. Cummings, "Application of the Fault Tree Technique to a Nuclear Reactor Containment System," (in Reliability and Fault Tree Analysis, R. E. Barlow, J. B. Fussell and N. D. Singpurwalla, Editors, SIAM 1975).

[14] E. P. Epler, "Common Mode Failure Considerations in the Design of Systems for Protection and Control," Nuclear Safety, 10 (1), Jan-Feb., 1969.

[15] C. A. Ericson, System Safety Analytical Technology - Fault Tree Analysis, Boeing Company, Seattle, Rept. D2-113072-2 (1970).

[16] C. A. Ericson, System Safety Analytical Technology - Preliminary Hazards Analysis, The Boeing Co., Seattle, Rept. D2-113072-1 (1969), pp. 13-16.

[17] J. D. Esary and F. Proschan, "A Reliability Bound for Systems of Maintained Interdependent Components," Journal of American Statistical Association, 65, pp. 329-338 (1970).

[18] J. D. Esary and F. Proschan, "Coherent Structures with Non-identical Components," Technometrics, 5, 191 (1963).

[19] J. D. Esary, F. Proschan and D. W. Walkup, "Association of Random Variables, with Applications," Ann. Math. Statist., 38, 1466, (1967).

[20] F. R. Farmer, "Reactor Safety and Siting: A Proposed Risk Criterion," Nuclear Safety, 8 (6), (1967).

[21] "Fault Tree Analysis - Apollo 13 Incident," The Boeing Co., for MSC Apollo 13 Review Board, Contract NAS-9-10364, (1970).

[22] J. B. Fussell, Aerojet Nuclear Co., Idaho Falls, private communication (1975).

[23] J. B. Fussell, "Computer Aided Fault Tree Construction for Electrical Systems," (in Reliability and Fault Tree Analysis, R. E. Barlow, J. B. Fussell and N. D. Singpurwalla, editors, SIAM, 1975).

[24] J. B. Fussell, Fault Tree Analysis - Concepts and Techniques (NATO Advanced Study Inst. on Generic Techniques of System Reliability Assessment, Liverpool, England, 1973).

[25] J. B. Fussell, "How to Hand-Calculate System Reliability Characteristics," IEEE Trans. on Rel., R-24, (3), (1975).

[26] J. B. Fussell, Aerojet Nuclear Co., Idaho Falls, private communication (1974).

[27] J. B. Fussell, Particularities of Fault Tree Analysis, Aerojet Nuclear Co., Idaho Falls, Report for Automation Industries, (1974).

[28] J. B. Fussell, Special Techniques for Fault Trees Analysis, Aerojet Nuclear Co., Idaho Falls, April, 1974.

[29] J. B. Fussell, Synthetic Tree Model - A Formal Methodology for Fault Tree Construction, Aerojet Nuclear Co., Idaho Falls, Rept. ANCR-1098 (1973).

[30] J. B. Fussell, et al.,"Fault Trees - A State of the Art Discussion,"IEEE Trans. on Reliability R-23, (1), (1974), pp. 51-55.

[31] J. B. Fussell, et al., MOCUS - A Computer Program to Obtain Minimal Sets, Aerojet Nuclear Co., Idaho Falls, (1974).

[32] W. C. Gangloff, An Evaluation of Anticipated Operational Transients in Westinghouse Pressurized Water Reactors, Westinghouse Electric Corporation, Pittsburgh, Rept, WACP-7486 (1971) p.2-4.

[33] B. J. Garrick, et al., Reliability Analysis of Nuclear Power Plant Protection Systems, Holmes & Narver, Inc., Los Angeles, Rept. HN 190, (1967).

[34] B. J. Garrick, Holmes & Narver, Inc., Los Angeles, private communication (1975).

[35] C. P. Gilmore, "Why Apollo 13," Pop Sci 197 (4), p. 64.

[36] A. E. Green and A. J. Bourne, Safety Assessment with Reference to Automatic Protective Systems for Nuclear Reactors, United Kingdom Atomic Energy Authority, AHSB (s) R 117, (1966).

[37] V. L. Grose, Systems Safety, Course 104, School of Continuing Engineering Education, George Washington University, Washington, D.C., 20006, (1974).

[38] D. F. Haasl, "Advanced Concepts in Fault Tree Analysis," in Proc. Systems Safety Symp., Univ. of Wash, and the Boeing Co., Seattle, (1965).

[39] D. F. Haasl, Fault Tree Construction Guide, Safety Engineering Division, Elgin Air Force Base, Florida, L/C F08635-75-C-0006, (1974).

[40] D. F. Haasl, Institute of Systems Sciences, Bellevue, Wash., private communication (1972).

[41] W. Hammer, Handbook of System and Product Safety (Prentice-Hall, Inc., Englewood Cliffs, NJ, 1972).

[42] W. L. Headington, M. E. Stewart, J. O. Zane, Fault Tree Analysis of the PBF Transient Rod Drive System, Phillips Petroleum Co., Idaho Falls, Rept. IDO-17274, (1968) pp. 139-140.

[43] E. Henley, Dept. of Chem. Engr., Univ. of Houston, private communication (1974).

[44] I. M. Jacobs, "Reliability of Engineered Safety Features as a Function of Testing Frequency," Nucl. Safety 9 (4), 303 (1968).

[45] W. E. Jordan, "Failure Modes, Effects, and Criticality Analysis," in Proc. Ann. Reliab. and Maintain. Sym. (San Francisco, 1972).

[46] J. Kielson, Markov Chain Models - Rarity and Exponentiality, Operations Research Center, Univ. of Calif., Berkeley, Rept. ORC 74-32 (1974).

[47] H. E. Lambert, Systems Safety Analysis and Fault Tree Analysis, Lawrence Livermore Laboratory, Livermore, Rept. UCID 16238 (1973).

[48] E. Levens, "Hazards Recognition - the Movement Ahead," Environmental Control Management, February (1970), p. 20.

[49] C. O. Miller, "Hazards Analysis and Identification in System Safety Engineering," In Proc. Reliability and Maintainability, (San Francisco, 1968), P. 342.

[50] MIL-STD-882, Military Standard, System Safety Program for Systems and Associated Subsystems and Equipment: Requirements for, Department of Defence, Wash., D.C., (1969).

[51] J. Murchland, "Fundamental Probability Relations for Repairable Items," NATO Advanced Study Institute on Generic Techniques in Systems Reliability Assessment, The University of Liverpool, July 17-27, 1973.

[52] J. Murchland and G. G. Weber, "A Moment Method for the Calculation of a Confidence Interval for the Failure Probability of a System," In Annual Reliability and Maintainability Symposium, (San Francisco, 1972), p. 562.

[53] W. E. McFatter, "Reliability Experiences in a Large Refinery," Chem. Engr. Progr. 68 (3), 52-55 (1972).

[54] A. McGibbon, SFAULTS, in Fault Tree Analysis Programs Available on the 7600/6600 Computers, Lawrence Livermore Laboratory, Rept. LER 73-100701, (1973).

[55] K. L. Nielsen, Methods in Numerical Analysis, (The MacMillan Company, New York, 1956).

[56] P. K. Pande, M. E. Spector and P. Chatterjee, Computerized Fault Tree Analysis: Tree and Micsup, Operations Research Center, Univ. of Calif.,Berkeley, Rept. ORC 75-3 (1975).

[57] G. J. Powers, et al., "Fault Tree Synthesis for Chemical Processes," AICHE Journal, 20 (2), 376-387 (1974).

[58] G. J. Powers, F. C. Tompkins and S. A. Lapp, "A Safety Simulation Language for Chemical Processes: A Procedure for Fault Tree Synthesis," In Reliability and Fault Analysis, R. E. Barlow, J. B. Fussell and N. D. Singpurwalla, editors, SIAM, 1975).

[59] N. H. Roberts, Mathematical Models in Reliability Engineering (McGraw-Hill Book Co., New York, 1964) p. 243.

[60] W. P. Rogers, Introduction to System Safety Engineering (John Wiley and Sons, Inc., New York, 1966).

[61] A Rosenthal, Dept. of Computer Science, Univ. of Mich., Ann Arbor, private communication (1975).

[62] S. Ross, Applied Probability Models with Optimization Applications (Holden-Day, San Francisco, 1970) pp. 31-84.

[63] S. Ross, Multicomponent Reliability Systems, Operations Research Center, Univ. of Calif., Berkeley, Rept. 74-4 (1974).

[64] S. Ross, On Time to First Failure in Multicomponent Exponential Reliability Systems, Operations Research Center, Univ. of Calif., Berkeley, Rept. ORC 74-8 (1974).

[65] G. H. Sandler, System Reliability Engineering (McGraw-Hill Book Co., New York, 1964), p. 243.

[66] D. Sethi, Maintained Systems, Random Replacement Policies, Operations Research Center, Univ. of Calif., Berkeley, to be published.

[67] SYREL Data Bank, Systems Reliability Service, UKAEA.

[68] System Safety Symposium, Proc, University of Washington and The Boeing Company (Seattle, 1965).

[69] United States Atomic Energy Commission, Appendix I, "Accident Definition and Use of Event Trees," in Reactor Safety Study, Rept. WASH 1400 (Draft), (1974).

[70] United States Atomic Energy Commission, Appendix II, PWR Fault Trees," in Reactor Safety Study, Rept. WASH 1400 (Draft), (1974).

[71] United States Atomic Energy Commission, Appendix II (Vol. 1), Fault Tree Methodology," in Reactor Safety Study, Rept. WASH 1400, (Draft), (1974).

[72] United States Atomic Energy Commission, "Appendix III, "Failure Data," in Reactor Safety Study, Rept. WASH 1400 (Draft), (1974).

[73] United States Atomic Energy Commission, "Appendix IV, Common Mode Failures," in Reactor Safety Study, Rept. WASH 1400 (Draft), (1974).

[74] United States Atomic Energy Commission, "Appendix V, "Quantitative Results of Accident Sequences," in Reactor Safety Study, Rept. WASH 1400 (Draft), (1974).

[75] United States Atomic Energy Commission, "Appendix V. Attch. 1, Source Term Evaluations for Postulated Accident Sequences," in Reactor Safety Study, Rept. WASH 1400 (Draft), (1974).

[76] United States Atomic Energy Commission, Evaluation of Nuclear Power Plant Availability, Office of Operations Evaluation, OOE-ES-001, (1974).

[77] United States Atomic Energy Commission, Reactor Safety Study, Rept. WASH 1400 (Draft), (1974).

[78] W. E. Vesely, Div. of Reactor Safety Research, Nuclear Regulatory Commission, Wash, D.C., private communication (1974).

[79] W. E. Vesely, "Quantitative Evaluations of a Fault Tree Phase 1: Component Characteristics" taken from System Safety Analysis and Fault Tree Analysis, course offered by D. F. Haasl, (1972).

[80] W. E. Vesely, "Reliability Quantification Techniques Used in the Rasmussen Study," (in Reliability and Fault Tree Analysis, R. E. Barlow, J. B. Fussell and N. D. Singpurwalla, editors, SIAM, 1975).

[81] W. E. Vesely, "A Time-Dependent Methodology for Fault-Tree Evaluation," Nucl. Engr. and Design (13), 337 (1970).

[82] W. E. Veseiy and R. E. Narum, PREP and KITT: Computer Codes for the Automatic Evaluation of Fault Trees, Idaho Nuclear Corp., Idaho Falls, Rept. IN 1349 (1970).

[83] Westinghouse Electric Corp., Guideline for the FMEA, Pittsburgh, WG 40194, (Internal Report) (1974).

[84] R. B. Worrell, Using the Set Equation Transformation System in Fault Tree Analysis, Sandia Laboratories, Albuquerque, NM, Rept. SAND 74-0240, (1974).

[85] E. S. Yoon, Fault Tree Analysis and its Applications for the Safety of Chemical Processing Systems (unpublished work), Dept. of Chem. Eng., MIT, Dec., 1973.

[86] J. Young and L. L. Conradi, "Including the Potential for Human Error in Fault Tree Analyses of Nuclear Power Systems," (In Proc. 2nd International System Safety Conference, San Diego, 1975).

[87] J. Young, "Using the Fault Tree Techniques," (in Reliability and Fault Tree Analysis, R. E. Barlow, J. B. Fussell and N. D. Singpurwalla, editors, SIAM, 1975).

APPENDIX A

IMPORTANCE COMPUTER CODE

The computer code, IMPORTANCE, computes various measures of probabilistic importance of basic events and cut sets to a fault tree. The code requires as input the minimal cut sets, the failure rates and the fault duration times (i.e., the repair times) of all basic events contained in the min cut sets. The failure and repair distributions are assumed to be exponential. The code can compute seven measures of basic event importance and two measures of cut set importance. All measures are computed assuming statistical independence of basic events.

The code allows seven measures of basic event importance and two measures of cut set importance to be computed. These are shown in Table A-1.

A.1 Rationale for Conditioning

As shown in the list of expressions, the measures that depend upon one point in time are conditioned on the system unavailability, $g(\underline{q}(t))$. The measures that are time integrated quantities depend upon the sequences of events leading to system failure. They are conditioned on the expected number of system failures, $E[N_s(t)]$. When repair is not allowed, $g(\underline{q}(t))$ is identically $E[N_s(t)]$. When repair is allowed, $g(\underline{q}(t))$ does not depend upon any previous system state as does $E[N_s(t)]$. The time integrated measures of importance when divided by $E[N_s(t)]$ approaches an asymptotic value for large time when all basic events have a finite fault duration time (i.e., all system components are repairable). For example,

TABLE A-1

Importance Measures Computed in IMPORTANCE Computer Code

Basic Event Measure	Expression
1. Birnbaum	$\frac{\partial g(\underline{q}(t))}{\partial q_i(t)} = g(1_i, \underline{q}(t)) - g(0_i, \underline{q}(t))$
2. Criticality	$\frac{\{g(1_i, \underline{q}(t)) - g(0_i, \underline{q}(t))\}\ q_i(t)}{g(\underline{q}(t))}$
3. Upgrading Function	$\frac{\alpha_i}{g(t, \underline{\alpha})} \cdot \frac{g(t, \underline{\alpha})}{\partial \alpha_i}$
4. Fussell-Vesely	$\frac{g_i(\underline{q}(t))}{g(\underline{q}(t))}$
5. Barlow-Proschan	$\frac{\int_0^t \{g(1_i, \underline{q}(t')) - g(0_i, \underline{q}(t'))\}\ dw_{f,i}(t'))}{E[N_s(t)]}$
6. Steady State BP	$\frac{[g(1_i, \underline{\bar{A}}) - g(0_i, \underline{\bar{A}})]/(\mu_i + \nu_i)}{\sum_{i=1}^{n}[g(1_j, \underline{\bar{A}}) - g(0_j, \underline{\bar{A}})]/(\mu_j + \nu_j)}$
7. Sequential Contributory	$\sum_{\substack{j \\ i\neq j \\ i\&j\in K_l}} \int_0^t \frac{\{g(1_i,1_j,\underline{q}(t')) - g(1_i,0_j,\underline{q}(t'))\}q_i(t')dw_{f,j}(t')}{E[N_s(t)]}$ for some l

Cut Set Measure	Expression
1. Barlow-Proschan	$\frac{\sum_{i\in K_j} \int_0^t [1 - g(0_i, \underline{1}^{K_j-\{i\}}, \underline{q}(t))] \prod_{\substack{j\neq i \\ j\in K_j}} q_j(t) dw_{f,i}(t)}{E[N_s(t)]}$
2. Fussell-Vesely	$\frac{\prod_{i\in K_j} q_i(t)}{g(\underline{q}(t))}$

the Barlow-Proschan measure of basic event importance approaches the asymptotic value given by the steady state B-P measure of importance.

A.2 Options to IMPORTANCE Computer Code

Four options are allowed in the use of the code. The first option, Option 1, computes measures of importance as a function of time. The input data required are the points in time for which the measures are to be computed. The basic event data, i.e., the failure rates and repair times, are expressed in time units (e.g., per hour and hours). The second and third options, Options 2 and 3, compute the measures of importance as a function of the probability of the top event. These options do not permit repair. The second option requires the failure rates to be given in time units. The third option allows failure rates to be expressed proportionally (i.e., assumption of proportaional hazards). These options also require as input the probabilities of the top event for which the measures are computed. The fourth option, Option 4, computes measures of importance as a function of a reference time unit, μ. The basic event data is given in terms of mean time to failure and mean repair times expressed in terms of the reference time unit μ.

The computer output consists of a series of tables listing the measures of importance in descending order as a function of the data input (i.e., time, probability of top event or time units of μ). There is also an option that generates data points suitable for plotting.

Data Input

First Card: TITLE (I), I = 1, 10 FORMAT (10A8)

The first card is the title card.

Second Card: IDATA, NTPT; FORMAT (2I10)

IDATA specifies one of the possible four options,

NTPT number of data points on the third card.

OPTION TABLE

IDATA	OPTION	DATA INPUT ON THIRD CARD
0 or 1*	1	REAL TIME
2	2	PROBABILITY OF TOP EVENT
3	3	PROBABILITY OF TOP EVENT
4	4	UNITS OF A REFERENCE TIME UNIT

*i.e., left blank

Third Card:

If IDATA = 2 or 3 PTOP(I) $1 \leq I \leq$ NTPT

If IDATA = 0, 1 or 4 TIME (I) $1 \leq I \leq$ NTPT

FORMAT (8E10.3)

Fourth Card: IX(I) I = 1, 7; FORMAT (7I10)

Basic Event Importance Options

If IX(I) = 1 Measure I computed

If IX(I) = 0 or blank Measure I not computed

OPTION TABLE

I	BASIC EVENT MEASURE
1	BIRNBAUM
2	CRITICALITY
3	UPGRADING FUNCTION
4	FUSSELL-VESELY
5	BARLOW PROSCHAN & STEADY STATE BP
6	CONTRIBUTORY

Fifth Card: IY(I) I = 1, 2; FORMAT (2I10)

Cut Set Importance Options:

If IY(I) = 1 Measure I Computed

If IY(I) = 0 or blank Measure I not computed

OPTION TABLE

I	CUT SET MEASURE
1	Fussell-Vesely
2	Barlow-Proschan

Sixth Card: IBPMX, IFVMX; FORMAT (2I10)

IBPMX and IFVMX specify the maximum order of the cut sets to be examined in the cut set options given on the fifth card. Card is left blank if cut set options are not invoked.

Seventh Card: IPLOT, FACTOR; FORMAT (I10, F10.5)

If IPLOT = 1 Data points suitable for plotting are generated for the measure options given on cards 3 and 4.

If IPLOT left blank Data points not generated

FACTOR is a number between 0 and 1. If XMAX represents the value of most important event (or cut set), then data points for basic events (or cut sets) with an importance value greater than XMAX*FACTOR are generated.

The data points are generated in pairs (X, Y). Where X represents the abscissa, time or probability of the top event and Y represents the importance value computed at X.

Eighth Card: NBE, NCS; FORMAT (2I10)

NBE is the number of basic events given in the basic event data.

NCS is the number of cut sets.

Cards 9 through 9+NBE-1: I, LAMDA(I), TAU(I), NAM(I)

FORMAT (I9, X, 2E10.3, 2X, A8)

Basic Event Data

I is the number designated to the basic event;

LAMDA(I) failure rate, prportional hazard rate, or mean time to failure expressed in units of μ.

TAU(I) repair time

NAM(I) alphanumeric designator for basic event I.

Restrictions on Data Input

Option	LAMDA	TAU
1	failure rate expressed in time units	repair time expressed in time units
2	failure rate expressed in time units	repair time must be 0 or left blank (convention indicating repair not allowed)
3	proportional hazard rate	repair time must be 0 or left blank
4	lamda in this case is not a failure rate but is the reciprocal, mean time to failure, expressed in units of μ.	expressed in units of μ.

To allow for houses and inhibit gates, the convention of Narum and Vesely in the PREP and KITT computer codes is adopted [A-1]. The following interpretations for LAMDA and TAU hold in all four options.

LAMDA (I)	TAU (I)	Interpretation
	equal to 0.0	basic event is a house that is turned off
negative or 0		
	equal to 1.0	basic event is a house that is turned on
	greater than 0.0 but less than 1.0	basic event is an inhibit gate, its probability of occurrence is TAU(I).

Further Restrictions on Basic Event Data

It is necessary that basic event data be placed numerically in order and numbered 1 through NBE. All basic events that appear in at least one cut set must be listed. Irrelevant basic events, i.e., basic events that do not appear in any cut set, may be listed. The code automatically eliminates irrelevant events.

Cards 9+NBE through 9+NBE+NCS-1; FORMAT (1615)

Cut Sets Data

Cut sets up to order 15 are accepted. It is necessary that the cut sets be placed in ascending order according to order, i.e., cut sets that contain one event be listed first, cut sets of order 2 be listed second, etc. The basic events contained in the cut sets appear as integer numbers 1 through the number NBE.

A.3 Sample Output of IMPORTANCE Computer Code

Figures A.1 through A.6 illustrate sample inputs and outputs for three options of the IMPORTANCE computer code. Sample inputs for options 1 through 3 are in Figures A.1, A.3, and A.5. In examining the sample

input for option 1 in Fig. A.1, the following line numbers give the indicated information:

Line Number	Information
2	Option 1 employed, 3 data points on Line 3
3	3 time points
4	Birnbaum's measure of basic event importance is to be computed
5	Barlow-Proschan measure of cut set importance to be computed.
6	maximum order of cut set importance to be computed is 5.
7	plot option to be invoked, FACTOR = .0001
8	number of basic events = 17; number of cut sets = 16.
9-25.	first three components repairable; remaining are unrepairable.
26-41.	There are 16 cut sets whose basic events are numbered 1 through 17.

Two output files are generated for option 1 in Figure A.2 Tables in Figure A.2 list for the measure indicated the importance value in descending order of each basic event or cut set as a function of mission time. The probability of the top event is also given as well as the expected number of system failures in the case of time integrated importances. Cut sets given in Fig. A.2 are indexed according to number. A reference table for the min cut sets is given. All basic events and cut sets whose importance value lies within the range FACTOR*XMAX are given as paired data points in the plotting file. The data points are located through the use of tables in Fig. A.2.

The output for option 2 in Fig. A.4 is basically the same as in option 1 except that importances are computed as a function of the probability of the top event. For option 3 as indicated in Fig. A.5, failure rates are expressed proportionally and computed as a function of the probability of the top event as shown in Fig. A.6. The input and output for option 4 is illustrated in Fig. D.3 of Appendix D. Note that the input in Fig. D.3, the mean time to failure and mean repair time, is expressed in units of μ. Observing the output in fig. D.3, we note that the steady state rate of system breakdown is computed with corresponding importances.* The steady rate of system breakdown is computed when $\lim_{t\to\infty} g(\underline{q}(t)) < 1$, i.e., all cut sets must contain at least one repairable component. Fig. C.4 of Appendix C illustrates how data points from the plotting file can be plotted to show the time dependent behavior of the most important events.

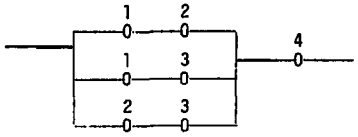
A.4 Programming Methods and Algorithms used in IMPORTANCE Code

All arrays of significant size are single string arrays. This provides the capacity of analyzing fault trees with a large number of cut sets with minimum wasted storage space.

The cut sets are read into the A array. The pointer array PTA locates cut sets according to order in the A array. The cut sets are rearranged according to the basic events contained in them with cut sets containing basic event 1 first, etc. The rearranged cut sets are placed in the B array. The pointer array PTB locates cut sets in the B array.

*The Barlow-Proschan measure basic event importance as a function of time is omitted in Fig. D.3.

As an example of the above process, consider the following reliability network diagram, a two-out-of-three structure in series with component 4.



The min cut sets are {4}, {1,2}, {1,3}, {2,3}. These cut sets are stored in the A array as follows, A → (4, 1, 2, 1, 3, 2, 3). The pointer array PTA in this case has the following values:

	I →				
PTA(I,J)	1	2	3 . . .	15	
J 1	1	2	0 . . .	0	← Locates where cut sets of Order I starts
↓ 2	1	7	0 . . .	0	← Cut sets of Order I ends
3	1	3	0 . . .	0	← Number of Cut Sets of Order I
4	1	4	0 . . .	0	← Number of Cut Sets of Order I and less

The B array is filled in the following manner

B → (2, 3, 2, 4, 3, 4, 1).

The pointer array PTB

PTB → (0, 2, 4, 6, 7)

I → 1, 2, 3, 4, 5.

To find what cut sets in the B array contain the basic event I, look at PTB(I) + 1 for the starting position in the B array, and PTB(I + 1) for the end position. For example, the PTB array tells us that cut sets that contain basic event 2 starts at position 3 and ends at position 4 in the B array.

The rationale for setting up the B array is that it allows for computational efficiency in computing the probability of the top event and Birnbaum's measure of importance (i.e., the partial derivative). The seven out of nine measures of importance are a function of Birnbaum's measure of importance.

Birnbaum's measure of importance and the probability of the top event are computed using the min cut upper bound. For reliable systems, i.e., probability of system failure less than .1, experience has shown that the min cut upper bound is an accurate approximation.

To minimize rounding error in the computation of the min cut upper bound, an algorithm suggested by Murchland and Weber [2] is used. The algorithm is understood if we consider the probability of the union of two statistically independent events b_1 and b_2, with probabilities P_1 and P_2,

$$P(b_1 \cup b_2) + P_1 + (1-P_1)P_2.$$

In taking the union of many statistically independent events, where P_i is a small quantity, it is best to use the above formula successively instead of

$$1 - \prod_i (1 - P_i).$$

The above is commonly given as the expression to compute the min cut upper bound with P_i as the probability of occurrence for cut set i and $P_i = \prod_{\ell \in K_i} q_\ell$ where q_ℓ is the probability of occurrence of basic event ℓ. To further increase the accuracy in computing the min cut upper bound, the algorithm starts with highest order cut sets first and then adds successively the cut sets in descending order. This eliminates some of the inaccuracy of

adding numbers that differ in orders of magnitude. In general, cut sets of higher order have a lower probability of occurrence.

To understand the method employed in calculating Birnbaum's measure of importance, $g(1_i, \underline{q}) - g(0_i, \underline{q})$, the following notation is introduced

INDICATOR VARIABLES

K^i union of all min cut sets containing i

$K^{\neq i}$ union of all min cut sets not containing i.

PROBABILITY EXPRESSIONS

$g(\underline{q}(t)) = P(TOP) =$ probability of Top Event.

$P_1(K^i)$ = probability that indicator variable $K^i = 1$ with $Y_i = 1$.

$P_0(K^i)$ = probability that indicator variable $K^i = 1$ with $Y_i = 0$.

The following relationship holds (assuming statistical independence of cut sets)

$$P(TOP) = P(K^i \cup K^{\neq i})$$
$$= P(K^i) + [1 - P(K^i)]\, P(K^{\neq i})$$

$$\frac{P(TOP) - P(K^i)}{1 - P(K^i)} = P(K^{\neq i}).$$

In terms of $P(K^{\neq i})$, we can generate Birnbaum's measure of importance

$$g(1_i, \underline{q}) = P(K^{\neq i} \cup K^i \,|_{y_i=1})$$
$$= P(K^{\neq i}) + [1 - P(K^{\neq i})]\, P_1(K^i)$$

similarly

$$g(0_i, \underline{q}) = P(K^{\neq i}) + [1 - P(K^{\neq i})]P_0(K^i),$$

however,

$$P_0(K^i) = 0.$$

Hence, $g(1_i, \underline{q}) - g(0_i, \underline{q}) = [1 - P(K^{\neq i})]\ [P_1(K^i)]$

by substitution

$$g(1_i, q) - g(0_i, q) = \frac{1 - P(TOP)}{1 - P(K^i)}\ P_1(K^i).$$

Hence, when the probability of the top event is known, computing the quantities $P(K^i)$ and $P_1(K^i)$ is sufficient to calculate Birnbaum's measure of importance. Setting up the B array enables us to locate the cut sets containing event i so that $P(K^i)$ and $P_1(K^i)$ can be calculated. By this method, we do not have to recompute the probability of the top event each time $g(1_i, \underline{q}) - g(0_i, \underline{q})$ is computed.

The above procedure in the code is accomplished by calling in order three subroutines, BEDATA(T, INTG), PTOPX, and BRNBAUM (IBE). The argument T in BEDATA represents one point in time for which the measures are to be computed. BEDATA then calculates $q_i(T)$ and $dw_{f,i}(t)$ (if INTG = 1) for each basic event. PTOPX then computes $P(K_j)$ for each cut set j, adds $P(K_j)$ successively in the manner previously described to calculate P(TOP). BRNBAUM (IBE) isolates the cut sets that contain basic event IBE and then calculates

$$\frac{1 - P(TOP)}{1 - P(K^{IBE})}\ P_1(K^{IBE}).$$

This procedure is demonstrated in the MAIN program, Fig. A.7, under the heading "Birnbaum's Measure of Basic Event Importance", line 148.

Simpson's rule of numerical integration is used in computing the time-integrated measures of importance. In the code, the ratio $P[TOP(T_2)]/P[TOP(T_1)]$ determines the number of integration points between $[T_1, T_2]$, $(T_1 \neq 0)$ provided that this ratio is between 10 and 100. Otherwise the minimum number of integration points is 10 and the maximum is 100. With

such smooth, well behaved expressions to be integrated and with such inaccuracies in the failure rate data, it was felt that to use a method more accurate than Simpson's rule is unjustified.

In the options where the importance measures are computed as a function of the probability of the top event, the mission time corresponding to the probability of the top event had to be found. Since $g(q(t))$ is a well behaved increasing function, Newton's method for approximating the roots of equations was employed.

The code sorts the output in descending order of importance. Of particular concern was the computation time required in sorting large arrays (i.e., a large number of basic events). The shell short is known to provide near maximum computational efficiency and is used in the code. In sorting arrays with N numbers, the shell sort requires approximately N log N steps to accomplish the sort.

Storage Requirements:

In analyzing large fault trees, it may be necessary to know the storage requirements in order that available core space is not exceeded. As mentioned previously the A and B arrays store the cut sets. These arrays must be dimensioned at least to $\sum_j I_{K_j}$ where I_{K_j} represents the order of cut set K_j. The sum is to be carried over all cut sets.

The C array is filled each time a measure of importance is computed. If there are NTPT time points, the C array is filled to the position, NTPT*NBE where NBE is the number of basic events. The D array stores the numbers corresponding to the basic events in the C array. The F array stores the rank of importance corresponding to the basic events in the C array. The D and F array have the same storage requirements as the C array. In the case where cut set importance is to be computed, the C, D

and E arrays must be dimensioned to the size NTPT*CSN where CSN is the total number of cut sets that are to be ranked. (The maximum order of cut sets to be ranked is specified on the sixth input card). Hence, the C, D, and F arrays must be dimensioned the maximum of the two quantities NTPT*NBE, NTPT*CSN.

The arrays

NAM, QBE, DELG, DF, ID

must be dimensioned NBE or greater.

The arrays

PTB, PTE

NBE+1 or greater

and the array

QCS

NCS or greater where NCS are the total number of cut sets.

```
 OPTION 1,  3 OUT OF 17 COMPONENTS REPAIRABLE,  PLOT OPTION INVOKED
      1
            3
1.000E+00 5.000E+00 1.000E+01
          1          0          0          0          0          0
          1          0
          5          0
          1     .00010
         17         16
          1 1.000E-03 1.▒▒▒▒+▒▒   COMP  1
          2 1.000E-04 1.▒▒▒▒+▒▒   COMP  2
          3 1.000E-05 1.▒▒▒▒+▒▒   COMP  3
          4 1.000E-03 0.▒▒▒▒+▒▒   COMP  4
          5 1.000E-03 0.▒▒▒▒+▒▒   COMP  5
          6 1.000E-03 0.▒▒▒▒+▒▒   COMP  6
          7 1.000E-03 0.▒▒▒▒+▒▒   COMP  7
          8 1.000E-04 0.▒▒▒▒+▒▒   COMP  8
          9 1.000E-05 0.▒▒▒▒+▒▒   COMP  9
         10 1.000E-03 0.▒▒▒▒+▒▒   COMP 10
         11 1.000E-03 0.▒▒▒▒+▒▒   COMP 11
         12 1.000E-03 0.▒▒▒▒+▒▒   COMP 12
         13 1.000E-03 0.▒▒▒▒+▒▒   COMP 13
         14 1.000E-04 0.▒▒▒▒+▒▒   COMP 14
         15 1.000E-03 0.▒▒▒▒+▒▒   COMP 15
         16 1.000E-03 0.▒▒▒▒+▒▒   COMP 16
         17 1.000E-03 0.▒▒▒▒+▒▒   COMP 17
      1
      2
      3
      4     5
      6     7
      8     9
      7    10
     10    15
     11    12
     12    13
     14    16
     15    17
      4     6    10
      4     9    12
      5     8    10
```

Fig. A.1 Input for Option 1

OPTION 1, 3 OUT OF 17 COMPONENTS REPAIRABLE, PLOT OPTION INVOKED

CROSS REFERENCE TABLE FOR PLOT OUTPUT

BIRNBAUM'S MEASURE OF BASIC EVENT IMPORTANCE

DATA PAIR RANGE 1 51

X COORINATE--TIME

BASIC EVENT	DATA PAIR RANGE	
COMP 4	1	3
COMP 5	4	6
COMP 6	7	9
COMP 7	10	12
COMP 8	13	15
COMP 9	16	18
COMP 10	19	21
COMP 11	22	24
COMP 12	25	27
COMP 13	28	30
COMP 14	31	33
COMP 15	34	36
COMP 16	37	39
COMP 17	40	42
COMP 1	43	45
COMP 2	46	48
COMP 3	49	51

OPTION 1, 3 OUT OF 17 COMPONENTS REPAIRABLE, PLOT OPTION INVOKED

BIRNBAUM'S MEASURE OF BASIC EVENT IMPORTANCE

PROB OF TOP EVENT=1.067E-03* PROB OF TOP EVENT=4.629E-03* PROB OF TOP EVENT=8.013E-03*
MISSION TIME=1.000E+00* MISSION TIME=5.000E+00* MISSION TIME=1.000E+01*

RANK	BASIC EVENT	IMPORTANCE*	RANK	BASIC EVENT	IMPORTANCE*	RANK	BASIC EVENT	IMPORTANCE*
1	COMP 1	9.999E-01*	1	COMP 1	9.993E-01*	1	COMP 1	9.989E-01*
2	COMP 2	9.998E-01*	2	COMP 2	9.959E-01*	2	COMP 2	9.929E-01*
3	COMP 3	9.969E-01*	3	COMP 3	9.954E-01*	3	COMP 3	9.920E-01*
4	COMP 10	1.698E-03*	4	COMP 10	9.954E-03*	4	COMP 10	1.984E-02*
5	COMP 12	1.426E-03*	5	COMP 12	9.909E-03*	5	COMP 12	1.965E-02*
6	COMP 7	1.594E-03*	6	COMP 7	9.905E-03*	6	COMP 7	1.965E-02*
6	COMP 15	1.996E-03*	6	COMP 15	9.905E-03*	6	COMP 15	1.965E-02*
7	COMP 6	1.110E-03*	7	COMP 6	5.554E-03*	7	COMP 6	1.116E-02*
8	COMP 4	9.994E-04*	8	COMP 4	4.999E-03*	8	COMP 4	9.970E-03*
9	COMP 9	9.994E-04*	9	COMP 9	4.989E-03*	9	COMP 9	9.966E-03*
10	COMP 5	9.994E-04*	10	COMP 5	4.989E-03*	10	COMP 5	9.969E-03*
11	COMP 17	9.984E-04*	11	COMP 17	4.965E-03*	11	COMP 17	9.871E-03*
11	COMP 11	9.984E-04*	11	COMP 11	4.965E-03*	11	COMP 11	9.871E-03*
11	COMP 13	9.984E-04*	11	COMP 13	4.965E-03*	11	COMP 13	9.871E-03*
12	COMP 8	9.984E-04*	12	COMP 8	4.964E-03*	12	COMP 8	9.871E-03*
12	COMP 14	9.984E-04*	12	COMP 14	4.964E-03*	12	COMP 14	9.871E-03*
13	COMP 16	9.999E-05*	13	COMP 16	4.976E-04*	13	COMP 16	9.915E-04*

Fig. A.2 Output for Option 1

OPTION 1, 3 OUT OF 17 COMPONENTS REPAIRABLE, PLOT OPTION INVOKED

CROSS REFERENCE TABLE FOR PLOT OUTPUT

BARLOW-PROSCHAN MEASURE OF CUT SET IMPORTANCE

DATA PAIR RANGE 52 87

X COORINATE--TIME

CUT SET NO. DATA PAIR RANGE

1	52	54
2	55	57
3	56	60
4	61	63
5	64	66
6	67	69
7	70	72
8	73	75
10	76	78
11	79	81
12	82	84
13	85	87

OPTION 1, 3 OUT OF 17 COMPONENTS REPAIRABLE, PLOT OPTION INVOKED

BARLOW-PROSCHAN MEASURE OF CUT SET IMPORTANCE

PROB OF TOP EVENT=1.067E-03*	PROB OF TOP EVENT=4.629E-03*	PROB OF TOP EVENT=8.013E-03*
MISSION TIME=1.000E+00*	MISSION TIME=5.000E+00*	MISSION TIME=1.000E+01*
EXP. NO. SYS FAIL=1.104E-03*	EXP. NO. SYS FAIL=5.703E-03*	EXP. NO. SYS FAIL=1.175E-02*

RANK	CUT SET NO	IMPORTANCE*	RANK	CUT SET NO	IMPORTANCE*	RANK	CUT SET NO	IMPORTANCE*
1	1	8.950E-01*	1	1	8.726E-01*	1	1	8.464E-01*
2	2	8.950E-02*	2	2	8.726E-02*	2	2	8.464E-02*
3	3	8.950E-03*	3	3	8.726E-03*	3	3	8.464E-03*
4	4	9.040E-04*	4	4	4.348E-03*	4	4	8.376E-03*
5	13	9.037E-04*	5	13	4.341E-03*	5	13	8.391E-03*
6	11	9.037E-04*	6	11	4.341E-03*	6	11	8.351E-03*
6	10	9.037E-04*	6	10	4.341E-03*	6	10	8.351E-03*
7	5	9.037E-04*	7	5	4.340E-03*	7	5	8.348E-03*
8	9	9.034E-04*	8	9	4.333E-03*	8	9	8.324E-03*
8	6	9.034E-04*	8	6	4.333E-03*	8	6	8.324E-03*
9	12	9.044E-05*	10	12	4.356E-04*	10	12	8.416E-04*
10	8	9.041E-05*	11	8	4.350E-04*	11	8	8.380E-04*
11	7	9.041E-06*	12	7	4.351E-05*	12	7	8.389E-05*
12	16	9.014E-07*	13	16	2.142E-05*	13	16	8.144E-05*
13	14	9.014E-07*	14	14	2.142E-05*	14	14	8.144E-05*
14	15	9.021E-09*	15	15	2.152E-07*	15	15	8.206E-07*

Fig. A.2 Cont'd

REFERENCE TABLE FOR MIN CUT SETS

CUT SET NO.	ORDER	BASIC EVENTS					
1	1	COMP 1					
2	1	COMP 2					
3	1	COMP 3					
4	2	COMP 4	COMP 5				
5	2	COMP 6	COMP 7				
6	2	COMP 6	COMP 8				
7	2	COMP 6	COMP 9				
8	2	COMP 7	COMP 10				
9	2	COMP 10	COMP 15				
10	2	COMP 11	COMP 12				
11	2	COMP 12	COMP 13				
12	2	COMP 14	COMP 16				
13	2	COMP 15	COMP 17				
14	3	COMP 4	COMP 6	COMP 11			
15	3	COMP 4	COMP 9	COMP 12			
16	3	COMP 5	COMP 6	COMP 10			

OUTPUT FOR PLOTTING OPTION, OPTION 1

1.000E+00	9.994E-04	5.000E+00	4.989E-03	1.000E+01	9.970E-03
1.000E+00	9.994E-04	5.000E+00	4.989E-03	1.000E+01	9.969E-03
1.000E+00	1.110E-03	5.000E+00	5.558E-03	1.000E+01	1.115E-02
1.000E+00	1.996E-03	5.000E+00	9.905E-03	1.000E+01	1.965E-02
1.000E+00	9.984E-04	5.000E+00	4.964E-03	1.000E+01	9.871E-03
1.000E+00	9.994E-04	5.000E+00	4.989E-03	1.000E+01	9.968E-03
1.000E+00	1.998E-03	5.000E+00	9.954E-03	1.000E+01	1.984E-02
1.000E+00	9.984E-04	5.000E+00	4.965E-03	1.000E+01	9.871E-03
1.000E+00	1.996E-03	5.000E+00	9.905E-03	1.000E+01	1.965E-02
1.000E+00	9.984E-04	5.000E+00	4.965E-03	1.000E+01	9.871E-03
1.000E+00	9.984E-04	5.000E+00	4.964E-03	1.000E+01	9.871E-03
1.000E+00	1.996E-03	5.000E+00	9.905E-03	1.000E+01	1.965E-02
1.000E+00	9.989E-05	5.000E+00	4.976E-04	1.000E+01	9.915E-04
1.000E+00	9.984E-04	5.000E+00	4.965E-03	1.000E+01	9.871E-03
1.000E+00	9.999E-01	5.000E+00	9.993E-01	1.000E+01	9.983E-01
1.000E+00	9.990E-01	5.000E+00	9.959E-01	1.000E+01	9.929E-01
1.000E+00	9.989E-01	5.000E+00	9.954E-01	1.000E+01	9.920E-01
1.000E+00	8.950E-01	5.000E+00	8.726E-01	1.000E+01	8.464E-01
1.000E+00	8.950E-02	5.000E+00	8.726E-02	1.000E+01	8.464E-02
1.000E+00	8.950E-03	5.000E+00	8.726E-03	1.000E+01	8.464E-03
1.000E+00	9.040E-04	5.000E+00	4.348E-03	1.000E+01	8.378E-03
1.000E+00	9.037E-04	5.000E+00	4.340E-03	1.000E+01	8.348E-03
1.000E+00	9.041E-05	5.000E+00	4.350E-04	1.000E+01	8.388E-04
1.000E+00	9.034E-04	5.000E+00	4.333E-03	1.000E+01	8.324E-03
1.000E+00	9.034E-04	5.000E+00	4.333E-03	1.000E+01	8.324E-03
1.000E+00	9.037E-04	5.000E+00	4.341E-03	1.000E+01	8.351E-03
1.000E+00	9.037E-04	5.000E+00	4.341E-03	1.000E+01	8.351E-03
1.000E+00	9.044E-05	5.000E+00	4.358E-04	1.000E+01	8.416E-04
1.000E+00	9.037E-04	5.000E+00	4.341E-03	1.000E+01	8.351E-03

Fig. A.2 Cont'd

```
 1  OPTION 2, NO REPAIR ALLOWED, PROB OF TOP EVENT INPUT
 2                2                3
 3 1.000E-05 5.000E-06 1.000E-05
 4           0         0         0          0          0          1
 5           0         0
 6
 7
 8          17        16
 9           1  1.000E-03 0.000E+00   COMP  1
10           2  1.000E-04 0.000E+00   COMP  2
11           3  1.000E-05 0.000E+00   COMP  3
12           4  1.000E-03 0.000E+00   COMP  4
13           5  1.000E-03 0.000E+00   COMP  5
14           6  1.000E-03 0.000E+00   COMP  6
15           7  1.000E-03 0.000E+00   COMP  7
16           8  1.000E-04 0.000E+00   COMP  8
17           9  1.000E-05 0.000E+00   COMP  9
18          10  1.000E-03 0.000E+00   COMP 10
19          11  1.000E-03 0.000E+00   COMP 11
20          12  1.000E-03 0.000E+00   COMP 12
21          13  1.000E-03 0.000E+00   COMP 13
22          14  1.000E-04 0.000E+00   COMP 14
23          15  1.000E-03 0.000E+00   COMP 15
24          16  1.000E-03 0.000E+00   COMP 16
25          17  1.000E-03 0.000E+00   COMP 17
26       1
27       2
28       3
29       4    5
30       6    7
31       8    9
32       6    9
33       7   10
34      10   15
35      11   12
36      12   13
37      14   16
38      15   17
39       4    6   10
40       4    9   12
41       5    6   10
```

Fig. A.3 Input for Option 2

1 OPTION 2, NO REPAIR ALLOWED, PROB OF TOP EVENT INPUT

SEQUENTIAL CONTRIBUTORY BASIC EVENT IMPORTANCE

PROB OF TOP EVENT=1.000E-06* MISSION TIME=9.009E-05*			PROB OF TOP EVENT=5.000E-06* MISSION TIME=4.504E-04*			PROB OF TOP EVENT=1.000E-05* MISSION TIME=9.009E-04*		
RANK	BASIC EVENT	IMPORTANCE*	RANK	BASIC EVENT	IMPORTANCE*	RANK	BASIC EVENT	IMPORTANCE*
1	COMP 10	5.116E-07*	1	COMP 10	3.842E-06*	1	COMP 10	7.872E-06*
2	COMP 12	5.116E-07*	2	COMP 12	3.842E-06*	2	COMP 12	7.872E-06*
3	COMP 7	5.116E-07*	3	COMP 7	3.842E-06*	3	COMP 7	7.872E-06*
4	COMP 15	5.116E-07*	4	COMP 15	3.842E-06*	4	COMP 15	7.872E-06*
5	COMP 6	4.505E-07*	5	COMP 6	2.132E-06*	5	COMP 6	4.369E-06*
6	COMP 4	4.058E-07*	6	COMP 4	1.921E-06*	6	COMP 4	3.936E-06*
7	COMP 5	4.058E-07*	7	COMP 5	1.921E-06*	7	COMP 5	3.936E-06*
8	COMP 11	4.058E-07*	8	COMP 11	1.921E-06*	8	COMP 11	3.936E-06*
9	COMP 17	4.058E-07*	9	COMP 17	1.921E-06*	9	COMP 17	3.936E-06*
9	COMP 13	4.058E-07*	9	COMP 13	1.921E-06*	9	COMP 13	3.936E-06*
10	COMP 16	4.058E-08*	10	COMP 16	1.921E-07*	10	COMP 16	3.936E-07*
11	COMP 8	4.058E-08*	11	COMP 8	1.921E-07*	11	COMP 8	3.936E-07*
12	COMP 14	4.058E-08*	12	COMP 14	1.921E-07*	12	COMP 14	3.936E-07*
13	COMP 9	4.058E-09*	13	COMP 9	1.921E-08*	13	COMP 9	3.936E-08*
14	COMP 3	.0E+00*	14	COMP 3	.0E+00*	14	COMP 3	.0E+00*
14	COMP 1	.0E+00*	14	COMP 1	.0E+00*	14	COMP 1	.0E+00*
14	COMP 2	.0E+00*	14	COMP 2	.0E+00*	14	COMP 2	.0E+00*

Fig. A.4 Output for Option 2

```
 1  OPTION 3, NO REPAIR ALLOWED, PROPORTIONAL HAZARDS, PROB OF TOP EVENT INPUT
 2                5                3
 3 1.000E-06 5.000E-05 1.000E-05
 4                0                0          1          0          0          0
 5                0                0
 6
 7
 8               17               17
 9                1  1.000E+01 0.000E+00  COMP  1
10                2  1.000E+00 0.000E+00  COMP  2
11                3  1.000E-01 0.000E+00  COMP  3
12                4  1.000E-01 0.000E+00  COMP  4
13                5  1.000E-01 0.000E+00  COMP  5
14                6  1.000E-01 0.000E+00  COMP  6
15                7  1.000E-01 0.000E+00  COMP  7
16                8  1.000E-02 0.000E+00  COMP  8
17                9  1.000E-01 0.000E+00  COMP  9
18               10  1.000E-01 0.000E+00  COMP 10
19               11  1.000E-01 0.000E+00  COMP 11
20               12  1.000E-01 0.000E+00  COMP 12
21               13  1.000E-01 0.000E+00  COMP 13
22               14  1.000E-02 0.000E+00  COMP 14
23               15  1.000E-01 0.000E+00  COMP 15
24               16  1.000E-01 0.000E+00  COMP 16
25               17  1.000E-01 0.000E+00  COMP 17
26     1
27     2
28     3
29     4     8
30     6     7
31     6     9
32     7     9
33     7    10
34    10    15
35    11    12
36    12    13
37    14    16
38    15    17
39     4     6    10
40     4     9    12
41     5     6    10
```

Fig. A.5 Input for Option 3

OPTION 3, NO REPAIR ALLOWED, PROPORTIONAL HAZARDS, PROB OF TOP EVENT INPUT
UPGRADING FUNCTION--BASIC EVENT IMPORTANCE

PROB OF TOP EVENT=1.000E-06*			PROB OF TOP EVENT=5.000E-06*			PROB OF TOP EVENT=1.000E-05*		
RANK	BASIC EVENT	IMPORTANCE*	RANK	BASIC EVENT	IMPORTANCE*	RANK	BASIC EVENT	IMPORTANCE*
1	COMP 1	9.009E-01*	1	COMP 1	9.009E-01*	1	COMP 1	9.008E-01*
2	COMP 2	9.009E-02*	2	COMP 2	9.009E-02*	2	COMP 2	9.008E-02*
3	COMP 3	9.009E-03*	3	COMP 3	9.009E-03*	3	COMP 3	9.008E-03*
4	COMP 10	1.623E-06*	4	COMP 10	8.116E-06*	4	COMP 10	1.623E-05*
5	COMP 12	1.623E-06*	5	COMP 12	8.116E-06*	5	COMP 12	1.623E-05*
6	COMP 7	1.623E-06*	6	COMP 7	8.116E-06*	6	COMP 7	1.623E-05*
6	COMP 15	1.623E-06*	6	COMP 15	8.116E-06*	6	COMP 15	1.623E-05*
7	COMP 6	9.009E-07*	7	COMP 6	4.504E-06*	7	COMP 6	9.008E-06*
8	COMP 4	8.116E-07*	8	COMP 4	4.058E-06*	8	COMP 4	8.115E-06*
9	COMP 5	8.116E-07*	9	COMP 5	4.058E-06*	9	COMP 5	8.115E-06*
10	COMP 11	8.116E-07*	10	COMP 11	4.058E-06*	10	COMP 11	8.115E-06*
10	COMP 13	8.116E-07*	10	COMP 13	4.058E-06*	10	COMP 13	8.115E-06*
10	COMP 17	8.116E-07*	10	COMP 17	4.058E-06*	10	COMP 17	8.115E-06*
11	COMP 8	8.116E-08*	11	COMP 8	4.058E-07*	11	COMP 8	8.115E-07*
11	COMP 14	8.116E-08*	11	COMP 14	4.058E-07*	11	COMP 14	8.115E-07*
12	COMP 16	8.116E-08*	12	COMP 16	4.058E-07*	12	COMP 16	8.115E-07*
13	COMP 9	8.116E-09*	13	COMP 9	4.058E-08*	13	COMP 9	8.115E-08*

Fig. A.6 Output for Option 3

```
      PROGRAM IMPORTX(CUTSETS,TAPE4=CUTSETS,OUTPT,TAPE5=OUTPT
     1,PLOT,TAPE6=PLOT)
C
CC ***************************************************************
CC
CC        WELCOME TO THE IMPORTANCE SMOGASBORD
CC
CC        THIS PROGRAM COMPUTES THE PROBABILISTIC IMPORTANCE
CC        OF BASIC EVENTS AND CUT SETS OF A FAULT TREE
CC        ACCORDING TO THE MEASURES GIVEN IN THE LAWRENCE
CC        LIVERMORE LABORATORY REPORT  UCRL 75853
CC
CC        IMPORTANCE PROGRAM WRITTEN BY HOWARD LAMBERT WHILE
CC        EMPLOYED AT LAWRENCE LIVERMORE LABORATORY AND A
CC        GRADUATE  STUDENT  AT UNIV. OF CALIF., BERKELEY
CC
C  ***************************************************************
C
      CALL DEVICE(6HCREATE,4HPLOT,10000)
      CALL DEVICE(6HCREATE,5HOUTPT,10000)
      CALL CHANGE(2H*HOPE)
      DIMENSION KLW(2),KUP(2),LCS(15),ENSF(8),BLNK(8),IX(8),IY(2)
      DIMENSION TITLE(10)
      DATA BLNK/8*1H /
      COMMON /UNIT/ LUN1,LUN2,LUN3
      COMMON /CB1/ A(1000),B(1000),PTA(15,4),PTB(101)
      COMMON /CB2/ LENGA,MAXORD,NBE,NCS,NIE,NIF,NRE
      COMMON /CB3/ C(1000),D(1000),F(1000),NAM
      COMMON /CB4/ QBE(100),QCS(100),PROBT,DEL       ,DF(100)
      COMMON /CB5/ LAMDA(100),TAU(100)
      COMMON /CB6/ NTPT,TIME(8),PTBP(8),IDATA,N...(8)
      COMMON /CB7/ ID(1000),E(1000),PTE(101)
      COMMON /CB8/ INDEX,IPRLW,FACTOR,IPR
      LUN1=4
      LUN2=5
      LUN3=6
      INTEGER A,B,D,E,F,PTB,PTE,PTA
      REAL LAMDA
C        READ TITLE
      READ(LUN1,1999)(TITLE(I),I=1,10)
C        READ CONTROL CARDS
CC
CC        SPECIFY OPTION AND NUMBER OF DATA POINTS
CC
      READ(LUN1,2000)IDATA,NTPT
      IF(IDATA.NE.2.AND.IDATA.NE.3)GO TO 1
      READ(LUN1,2001)(PTBP(I),I=1,NTPT)
      CALL SORT(PTBP,NTPT)
      IOPT=1
      GO TO 2
    1 READ(LUN1,2001)(TIME(I),I=1,NTPT)
      CALL SORT(TIME,NTPT)
      IOPT=0
    2 CONTINUE
C  ***************************************************************
C        BASIC EVENT IMPORTANCE OPTIONS
C
      READ(LUN1,2003)(IX(I),I=1,6)
C  ***************************************************************
C        CUT SET IMPORTANCE OPTIONS
```

Fig. A.7 IMPORTANCE Computer Code Listing

```
      C
            READ(LUN1,2000)(IY(I),I=1,2)
      C ***************************************************************
      C        SPECIFY MAXIMUM ORDER FOR CUT SET IMPORTANCE
      C
            READ(LUN1,2000)IBPMX,IFVMX
      C ***************************************************************
      C        PLOTTING OPTION
      C
            READ(LUN1,2002)IPLOT,FACTOR
      C ***************************************************************
      C        READ NUMBER OF BASIC EVENTS AND NUMBER OF MIN CUT SETS
      C
            READ(LUN1,2000)NBE,NCS
      C ***************************************************************
      C     READ IN COMPONENT DATA AND ALPHANUMERIC DESIGNATORS
      C
      C
            READ(LUN1,2005)(LAMDA(K),TAU(K),NAM(K),K=1,NBE)
      C ***************************************************************
      C        READ CUT SETS INTO A ARRAY
      C
            CALL READCS
      C
      C        ORDER CUT SETS ACCORDING TO BASIC EVENTS CONTAINED
      C        IN THEM -- STORE ORDERED CUT SETS IN B ARRAY
      C
            CALL CSARRAY
      C
      C        NORMALIZE DATA IF PROPORTIONAL HAZARDS SPECIFIED
      C
            IF(IOPT.EQ.0)GO TO 5
            IF(NIFDT.EQ.NIE) GO TO 6
            WRITE(LUN2,2006)
            CALL EXIT(1)
        6   CONTINUE
            IF(IDATA.EQ.1)GO TO 400
            XMAXL=LAMDA(1)
            DO 3 I=2,NIE
            IF(XMAXL.LT.LAMDA(I))XMAXL=LAMDA(I)
        3   CONTINUE
            DO 4 I=1,NIE
            LAMDA(I)=(1.E-2/XMAXL)*LAMDA(I)
        4   CONTINUE
      400   CONTINUE
      C
      C        GET TIME POINTS CORRESPONDING TO PROBABILITY
      C        OF TOP EVENT FOR OPTIONS 2 AND 3
      C
            CALL TRSDT
      C
            CONTINUE
            IF(IDATA.NE.4)GO TO 700
            DO 500 I=1,NIE
            LAMDA(I)=1./LAMDA(I)
      500   CONTINUE
      700   CONTINUE
            IXI=0
            DO 7 I=1,6
            IXI=IXI+IX(I)
        7   CONTINUE
            IF(IXI.EQ.0)GO TO 79
            INPG=NIE/40
            AN=NIE
            R= AN/40. - INPG
            IF(R.GT.0.)INPG=INPG + 1
            KLW(1)=1
            INDEX=NIE
            IF(NTPT.GT.4)GO TO 8
            KUP(1)= NTPT
            GO TO 9
        8   CONTINUE
            KUP(1)=4
            KLW(2)=5
            KUP(2)=NTPT
        9   CONTINUE
            IPRLW=1
            DO 401 I=1,NTPT
            T=TIME(I)
            CALL BEDATA(T,0)
            CALL PTOPX
            PTOP(I)=PROBT
      401   CONTINUE
            IF(IX(1).NE.1)GO TO 19
      C
      C ***************************************************************
      C
      C        BIRNBAUM'S MEASURE OF BASIC EVENT IMPORTANCE
      C
            IC=0
```

```
      DO 10 I=1,NTPT
      T=TIME(I)
      CALL BEDATA(T,0)
      CALL PTOPX
      DO 11 J=1,NIE
      IC=IC+1
      CALL BRNBAUM(J)
      C(IC)=DELG(J)
      D(IC)=J
 11   CONTINUE
 10   CONTINUE
      IF(IX(1).NE.1)GO TO 19
      DO 12 I=1,NTPT
      IUP=I*NIE
      ILW = NIE*(I-1) +1
      CALL PATSRT(C,D,F,ILW,IUP)
 12   CONTINUE
      LOCATE=1
      GO TO 100
 19   CONTINUE
C
C *********-**********-**********-**********-**********-**********
C
C          CRITICALITY BASIC EVENT IMPORTANCE
C
      IF(IX(2).NE.1)GO TO 29
      IC=0
      DO 20 I=1,NTPT
      T=TIME(I)
      CALL BEDATA(T,0)
      CALL PTOPX
      PTOP(I)=PROBT
      DO 21 J=1,NIE
      IC=IC+1
      D(IC)=J
      CALL BRNBAUM(J)
      C(IC)=DELG(J)*QBE(J)/PTOP(I)
 21   CONTINUE
 20   CONTINUE
      DO 22 I=1,NTPT
      IUP = I*NIE
      ILW=NIE*(I - 1)+1
      CALL PATSRT(C,D,F,ILW,IUP)
 22   CONTINUE
      LOCATE = 2
      GO TO 100
 29   CONTINUE
C
C ***********-**********-**********-**********-**********-**********
C
C          UPGRADING FUNCTION--BASIC EVENT IMPORTANCE
C
      IF(NIFDT.NE.NIE)GO TO 39
      IF(IX(3).NE.1)GO TO 39
      IC = 0
 32   CONTINUE
      DO 33 I=1,NTPT
      T=TIME(I)
      CALL BEDATA(T,0)
      CALL PTOPX
      PTOP(I)=PROBT
      DO 34 J=1,NIE
      IC=IC+1
      D(IC)=J
      CALL BRNBAUM(J)
      C(IC)=DELG(J)*LAMDA(J)*T*EXP(-LAMDA(J)*T)/PTOP(I)
 34   CONTINUE
 33   CONTINUE
 35   CONTINUE
      DO 36 I=1,NTPT
      IUP=I*NIE
      ILW= NIE*(I - 1) +1
      CALL PATSRT(C,D,F,ILW,IUP)
 36   CONTINUE
      LOCATE = 3
      GO TO 100
 39   CONTINUE
C
C ***********-**********-**********-**********-**********-**********
C
C          FUSSELL-VESELY BASIC EVENT IMPORTANCE
C
      IF(IX(4).NE.1)GO TO 49
      IC=0
      DO 40 I=1,NTPT
      CALL FVBE(I)
      PTOP(I)=PROBT
      DO 41 J=1,NIE
      IC=IC+1
      D(IC)=J
```

```
241   41    CONTINUE
242   40    CONTINUE
243         DO 42 I=1,NTPT
244         IUP=I*NIE
245         ILW= NIE*(I - 1)+1
246         CALL PATSRT(C,D,F,ILW,IUP)
247   42    CONTINUE
248         LOCATE= 4
249         GO TO 100
250   49    CONTINUE
251 C
252 C *****************************************************************
253 C
254 C           BARLOW-PROSCHAN BASIC EVENT IMPORTANCE
255 C
256         II=IX(5)+IX(6)+IY(1)
257         IF(II.EQ.0)GO TO 69
258         CALL POINTS
259         IC=0
260         DO 50 I=1,NTPT
261         DO 51 J=1,NIE
262         IC=IC+1
263         C(IC)=0.
264         D(IC)=J
265   51    CONTINUE
266   50    CONTINUE
267         DO 52 I=1,NTPT
268         CALL BPBE(I)
269   52    CONTINUE
270         IF(NIFDT.EQ.NIE)GO TO 55
271         IC=0
272         DO 53 I=1,NTPT
273         ENSF(I)=0.
274         DO 54 J=1,NIE
275         IC=IC+1
276         ENSF(I)=ENSF(I)+C(IC)
277   54    CONTINUE
278   53    CONTINUE
279         GO TO 57
280   55    DO 56 I=1,NTPT
281         ENSF(I)=PTOP(I)
282   56    CONTINUE
283   57    CONTINUE
284         IF(IX(5).NE.1)GO TO 69
285         IC=0
286         DO 58 I= 1,NTPT
287         DO 590 J= 1,NIE
288         IC=IC+1
289         C(IC)=C(IC)/ENSF(I)
290  590     CONTINUE
291         IUP=I*NIE
292         ILW= NIE*(I-1)+1
293         CALL PATSRT(C,D,F,ILW,IUP)
294   58    CONTINUE
295         LOCATE=5
296         GO TO 100
297   59    CONTINUE
298 C
299 C           STEADY STATE RATE OF BREAKDOWN
300 C
301         CALL SSBP(SSRBD,IERR)
302         IF(IERR.EQ.1)GO TO 69
303         WRITE(LUN2,3000)(TITLE(I),I=1,10)
304         WRITE(LUN2,3009)
305         WRITE(LUN2,3044)SSRBD
306         WRITE(LUN2,3046)PROBT
307         WRITE(LUN2,3041)BLNK(1)
308         WRITE(LUN2,3043)
309         DO 61 I=1,NIE
310         IBE=D(I)
311         WRITE(LUN2,3051)F(I),NAM(IBE),C(I)
312   61    CONTINUE
313   69    CONTINUE
314 C
315 C *****************************************************************
316 C
317 C           SEQUENTIAL CONTRIBUTORY BASIC EVENT IMPORTANCE
318 C
319         IF(IX(6).NE.1)GO TO 79
320         IC=0
321         CALL EARRAY
322         DO 70 I=1,NTPT
323         DO 71 J=1,NIE
324         IC=IC+1
325         C(IC)=0.
326         D(IC)=J
327   71    CONTINUE
328   70    CONTINUE
329         DO 72 I=1,NTPT
330         CALL CONTRIB(I)
```

```
331   72    CONTINUE
332         IC=0
333         DO 73 I=1,NTPT
334         DO 74 J=1,NIE
335         IC=IC+1
336         C(IC)=C(IC)/ENSF(I)
337   74    CONTINUE
338         IUP=I*NIE
339         ILW=NIE*(I -1)+1
340          CALL PATSRT(C,D,F,ILW,IUP)
341   73    CONTINUE
342         LOCATE = 6
343         GO TO 100
344   79    CONTINUE
345 C
346 C *************************************************************
347 C
348 C         BARLOW-PROSCHAN MEASURE OF CUT SET IMPORTANCE
349 C
350         IF(IY(1).NE.1)GO TO 89
351         J=NIE*NTPT
352         DO 83 I=1,J
353         C(I)=0.
354  83     CONTINUE
355         DO 80 I=1,NTPT
356         CALL BPCS(IBPMX,I,INCS)
357   80    CONTINUE
358         IC=0
359         DO 81 I=1,NTPT
360         DO 82 J=1,INCS
361         IC=IC+1
362         C(IC)=C(IC)/ENSF(I)
363         D(IC)=J
364   82    CONTINUE
365         IUP=I*INCS
366         ILW=INCS*(I-1)+1
367          CALL PATSRT(C,D,F,ILW,IUP)
368   81    CONTINUE
369         INPG=INCS/40
370         AN=INCS
371         R= AN/40. - INPG
372         IF(R.GT.0.)INPG=INPG+1
373         LOCATE=7
374         INDEX=INCS
375         GO TO 100
376   89    CONTINUE
377 C
378 C *************************************************************
379 C
380 C         FUSSELL-VESELY MEASURE OF CUT SET IMPORTANCE
381 C
382         IF(IY(2).NE.1)GO TO 99
383         J=IFVMX+1
384         DO 90 I=1,IFVMX
385         J=J-1
386        *IF(PTA(J,1).NE.0)GO TO 91
387   90    CONTINUE
388   91    ICSFV=J
389         IFVNCS=PTA(J,4)
390         IC=0
391         DO 92 I=1,NTPT
392         T=TIME(I)
393         CALL BEDATA(T,0)
394         CALL PTOPX
395         DO 93 J=1,IFVNCS
396         IC=IC+1
397         C(IC)=QCS(J)/PTOP(I)
398         D(IC)=J
399   93    CONTINUE
400         IUP= I*IFVNCS
401         ILW= IFVNCS*(I-1)+1
402          CALL PATSRT(C,D,F,ILW,IUP)
403   92    CONTINUE
404         INPG= IFVNCS/40
405         AN= IFVNCS
406         R= AN/40. - INPG
407         IF(R.GT.0.)INPG=INPG +1
408         LOCATE = 8
409         INDEX=IFVNCS
410         GO TO 100
411   99    CONTINUE
412         GO TO 1320
413  100    CONTINUE
414 C *************************************************************
415 C         WRITE OUTPUT
416         IF(IPLOT.EQ.1)GO TO 4000
417 1001    CONTINUE
418         CALL SWAPN
419         IADD=0
420         DO 121 I=1,2
```

```
      DO 120 J=1,INPG
      LUP=J*45 + IADD
      LLW=LUP-44
      IF(J.EQ.INPG)LUP=INDEX+IADD
      WRITE(LUN2,3000)(TITLE(I),I=1,10)
      GO TO(101,102,103,104,105,106,107,108)LOCATE
  101 WRITE(LUN2,3001)
      GO TO 109
  102 WRITE(LUN2,3002)
      GO TO 109
  103 WRITE(LUN2,3003)
      GO TO 109
  104 WRITE(LUN2,3004)
      GO TO 109
  105 WRITE(LUN2,3005)
      GO TO 109
  106 WRITE(LUN2,3006)
      GO TO 109
  107 WRITE(LUN2,3007)
      GO TO 109
  108 WRITE(LUN2,3008)
  109 CONTINUE
      KUPI=KUP(I)
      KLWI=KLW(I)
      ICOUNT=KUPI-KLWI+1
      WRITE(LUN2,3011)(BLNK(K),PTOP(K),K=KLWI,KUPI)
      IF(IDATA.NE.3)WRITE(LUN2,3021)(BLNK(K),TIME(K),K=KLWI,KUPI)
      IF((LOCATE.GE.5.AND.LOCATE.LE.7).AND.(NIFDT.NE.NIE))
     1WRITE(LUN2,3031)(BLNK(K),ENSF(K),K=KLWI,KUPI)
      IF(LOCATE.GT.6)GO TO 115
      WRITE(LUN2,3041)(BLNK(K),K=KLWI,KUPI)
      WRITE(LUN2,3043)
      GO TO(1110,1120,1130,1140)ICOUNT
 1110 CONTINUE
      DO 111  K=LLW,LUP
      IBE1=D(K)
      WRITE(LUN2,3051)F(K),NAM(IBE1),C(K)
  111 CONTINUE
      GO TO 120
 1120 CONTINUE
      DO 112  K=LLW,LUP
      IBE1=D(K)
      K1=K+INDEX
      IBE2=D(K1)
      WRITE(LUN2,3051)F(K),NAM(IBE1),C(K),F(K1),NAM(IBE2),C(K1)
  112 CONTINUE
      GO TO 120
 1130 CONTINUE
      DO 113  K=LLW,LUP
      IBE1=D(K)
      K1=K+INDEX
      K2=K+2*INDEX
      IBE2=D(K1)
      IBE3=D(K2)
      WRITE(LUN2,3051)F(K),NAM(IBE1),C(K),F(K1),NAM(IBE2),C(K1),F(K2),
     1NAM(IBE3),C(K2)
  113 CONTINUE
      GO TO 120
 1140 CONTINUE
      DO 114  K=LLW,LUP
      IBE1=D(K)
      K1=K+INDEX
      K2=K+2*INDEX
      K3=K+3*INDEX
      IBE2=D(K1)
      IBE3=D(K2)
      IBE4=D(K3)
      WRITE(LUN2,3051)F(K),NAM(IBE1),C(K),F(K1),NAM(IBE2),C(K1),F(K2),
     1NAM(IBE3),C(K2),F(K3),NAM(IBE4),C(K3)
  114 CONTINUE
      GO TO 120
  115 CONTINUE
      WRITE(LUN2,3042)(BLNK(K),K=KLWI,KUPI)
      WRITE(LUN2,3043)
      DO 116 K=LLW,LUP
      IDUM=K-INDEX
      WRITE(LUN2,3060)(F(IDUM+N*INDEX),D(IDUM+N*INDEX),C(IDUM+N*
     1INDEX),N=1,ICOUNT)
  116 CONTINUE
  120 CONTINUE
      IF(NTPT.LE.4)GO TO 1310
      IADD=4*INDEX
  121 CONTINUE
  130 CONTINUE
 1310 CONTINUE
      GO TO (19,29,39,49,59,79,89,99)LOCATE
 1320 CONTINUE
      IF(IY(1).NE.1.AND.IY(2).NE.1)GO TO 140
C
```

```
C ***********************************************************
C       WRITE OUT REFERENCE TABLE FOR CUT SETS
C
        IREF=IFVMX
        IF(IBPMX.GT.IFVMX)IREF=IBPMX
        WRITE(LUN2,3065)
        KNCS=0
        ILW=1
        IUP=0
        DO 131 I=1,IREF
        IF(PTA(I,1).EQ.0)GO TO 131
        INCS=PTA(I,3)
        DO 132 J=1,INCS
        IUP=IUP+1
        KNCS=KNCS+1
        KK=0
        DO 133 K=ILW,IUP
        JJ=A(K)
        KK=KK+1
        LCS(KK)=NAM(JJ)
  133   CONTINUE
        WRITE(LUN2,3070)(KNCS,I,(LCS(J1),J1=1,I))
        ILW=IUP +1
  132   CONTINUE
  131   CONTINUE
  140   CONTINUE
 1999   FORMAT(10A8//)
 2000   FORMAT(2I10)
 2001   FORMAT(8(E9.3,1X))
 2002   FORMAT(I10,F10.5)
 2003   FORMAT(6I10)
 2005   FORMAT(10X,2E10.3,2X,A8)
 2006   FORMAT( /10X,45HALL BASIC EVENTS OTHER THAN HOUSES AND INHIBIT,//
     110X,38HGATES MUST HAVE ZERO (.0) REPAIR TIMES,//10X,18HFOR OPTIONS
     22 AND 3)
C       TITLES
 3000   FORMAT(1H1,9X,10A8/)
 3001   FORMAT(10X,44HBIRNBAUM'S MEASURE OF BASIC EVENT IMPORTANCE,//)
 3002   FORMAT(10X,34HCRITICALITY BASIC EVENT IMPORTANCE,//)
 3003   FORMAT(10X,42HUPGRADING FUNCTION--BASIC EVENT IMPORTANCE,//)
 3004   FORMAT(10X,46HFUSSELL-VESELY MEASURE OF BASIC EVENT IMPORTANCE
     1,//)
 3005   FORMAT(10X,49HBARLOW-PROSCHAN MEASURE OF BASIC EVENT IMPORTANCE,
     1//)
 3006   FORMAT(10X,46HSEQUENTIAL CONTRIBUTORY BASIC EVENT IMPORTANCE,//)
 3007   FORMAT(10X,45HBARLOW-PROSCHAN MEASURE OF CUT SET IMPORTANCE,//
     1)
 3008   FORMAT(10X,44HFUSSELL-VESELY MEASURE OF CUT SET IMPORTANCE,//)
 3009   FORMAT(10X,45HSTEADY STATE BREAKDOWN BASIC EVENT IMPORTANCE,//)
 3011   FORMAT(4(1X,A1,18HPROB OF TOP EVENT=,E9.3,1H*))
 3021   FORMAT(4(1X,A1,18H      MISSION TIME=,E9.3,1H*))
 3031   FORMAT(/4(1X,A1,18HEXP. NO. SYS FAIL=,E9.3,1H*))
 3041   FORMAT(//4(1X,A1,28HRANK BASIC EVENT IMPORTANCE*))
 3042   FORMAT(///4(X,A1,28HRANK CUT SET NO  IMPORTANCE*))
 3043   FORMAT(/)
 3044   FORMAT(10X,41HRATE OF SYSTEM BREAKDOWN AT STEADY STATE=,E10.3///)
 3046   FORMAT(10X,31HLIMITING SYSTEM UNAVAILIBILITY=,E10.3)
 3051   FORMAT(4(2X,I4,2X,A8,3X,E10.3,1H*))
 3060   FORMAT(4(2X,I4,5X,I4,4X,E10.3,1H*))
 3065   FORMAT(1H1,8X,32HREFERENCE TABLE FOR MIN CUT SETS,///9X,33HCUT SET
     1 NO.  ORDER   BASIC EVENTS//)
 3070   FORMAT(13X,I4,6X,I2,5X,A 9(2X,A8)/30X,6(2X,A8))
 3100   FORMAT(10X,15HDATA PAIR RANGE,215//)
 3110   FORMAT(10X,17HX COORINATE--TIME,//)
 3120   FORMAT(10X,25HX COORDINATE--UNITS OF MU,//)
 3130   FORMAT(10X,38HX COORDINATE--PROBABILITY OF TOP EVENT,//)
 3150   FORMAT(10X,37HCROSS REFERENCE TABLE FOR PLOT OUTPUT,//)
 3200   FORMAT(10X,23HBASIC EVENT  DATA PAIR RANGE,//)
 3201   FORMAT(10X,28HCUT SET NO.  DATA PAIR RANGE,//)
 3300   FORMAT(12X,A8,5X,215)
 3301   FORMAT(14X,I6,5X,215)
 3999   FORMAT(6E11.3)
        GO TO 5000
C
C           PLOTTING OPTION OUTPUT
C
 4000   CONTINUE
        CALL PLOTS
        WRITE(LUN2,3000)(TITLE(I),I=1,10)
        WRITE(LUN2,3150)
        GO TO(4101,4102,4103,4104,4105,4106,4107,4108)LOCATE
 4101   WRITE(LUN2,3001)
        GO TO 4109
 4102   WRITE(LUN2,3002)
        GO TO 4109
 4103   WRITE(LUN2,3003)
        GO TO 4109
 4104   WRITE(LUN2,3004)
        GO TO 4109
 4105   WRITE(LUN2,3005)
        GO TO 4109
```

```
      WRITE(LUN2,3006)
      GO TO 4109
4107  WRITE(LUN2,3007)
      GO TO 4109
4108  WRITE(LUN2,3008)
4109   CONTINUE
      IPRUP=IPRLW+IPR*NTPT-1
      WRITE(LUN2,3100)IPRLW,IPRUP
      IF(IOPT.EQ.1)GO TO 4009
      IF(IDATA.NE.4)WRITE(LUN2,3110)
      IF(IDATA.EQ.4)WRITE(LUN2,3120)
      IF(LOCATE.LT.7)WRITE(LUN2,3200)
      IF(LOCATE.GT.6)WRITE(LUN2,3201)
      DO 4005 I=1,IPR
      L=ID(I)
      IDUM=L-INDEX
      WRITE(LUN3,3999)(TIME(N),C(IDUM+N*INDEX),N=1,NTPT)
      J1=IPRLW
      IPRLW=IPRLW+NTPT-1
      IF(LOCATE.LT.7)WRITE(LUN2,3300)NAM(L),J1,IPRLW
      IF(LOCATE.GT.6)WRITE(LUN2,3301)L,J1,IPRLW
      IPRLW=IPRLW+1
4005  CONTINUE
      GO TO 4010
4009  CONTINUE
      WRITE(LUN2,3130)
      IF(LOCATE.LT.7)WRITE(LUN2,3200)
      IF(LOCATE.GT.6)WRITE(LUN2,3201)
      DO 4006 I=1,IPR
      L=ID(I)
      IDUM=L-INDEX
      WRITE(LUN3,3999)(PTOP(I),C(IDUM+N*INDEX),N=1,NTPT)
      J1=IPRLW
      IPRLW=IPRLW+NTPT-1
      IF(LOCATE.LT.7)WRITE(LUN2,3300)NAM(L),J1,IPRLW
      IF(LOCATE.GT.6)WRITE(LUN2,3301)L,J1,IPRLW
      IPRLW=IPRLW+1
4006  CONTINUE
4010  CONTINUE
      GO TO 5001
5000  CONTINUE
      CALL EXIT(1)
      END
C ***********************************************************************
      SUBROUTINE READCS
C READCS STORES CUT SETS IN A ARRAY
      COMMON /UNIT/ LUN1,LUN2,LUN3
      COMMON /CB1/ A(1000),B(1000),PTA(15,4),PTB(101)
      COMMON /CB2/ LENGA,MAXORD,NBE,NCS,NIE,NIFDT,NRE
       INTEGER A,B,D,E,F,PTB,PTA
      DIMENSION ICS(16)
      DO 1 I=1,15
      PTA(I,1)=0
      PTA(I,2)=0
      PTA(I,3)=0
      PTA(I,4)=NCS+1
1     CONTINUE
      I1=0
      I2 = 1
      I3 = 1
      I4 = 0
      DO 2  ,=1,NCS
      READ(LUN1,1000)(ICS(K),K=1,16)
      IF(ICS(I2).EQ.0)GO TO 5
      K1= I2 + 1
      DO 3 J=K1,16
      I2=I2+1
      I1=I1+1
      IF(ICS(I2).EQ.0)GO TO 4
3     CONTINUE
4     CONTINUE
      PTA(I1,1)=I4 +1
      PTA(I3,2)=I4
      I3=I1
5     CONTINUE
      DO 6 J=1,I1
      I4=I4+1
      A(I4)=ICS(J)
6     CONTINUE
2     CONTINUE
      PTA(I1,2)=I4
C PTA(I,3),PTA(I,4) ARRAY
      I1=0
      DO 7 I=1,15
      IF(PTA(I,1).EQ.0)GO TO 7
      PTA(I,3) = (PTA(I,2)-PTA(I,1) +1)/I
      PTA(I,4) = I1 + PTA(I,3)
      I1=I1 + PTA(I,3)
7     CONTINUE
1000  FORMAT(16I5)
```

```
691         LENGA=I4
692         K1=16
693         DO 8 I=1,15
694         K1=K1-1
695         IF(PTA(K1,1).GT.0)GO TO 9
696       8 CONTINUE
697       9 MAXORD =K1
698         RETURN
699         END
700       C ***************************************************************
701         SUBROUTINE CSARRAY
702       C CSARRAY LOCATES BASIC EVENTS IN CUT SETS AND SETS
703       C UP THE B ARRAY.  PTB IS A POINTER ARRAY THAT LOCATES
704       C BASIC EVENTS IN THE B ARRAY
705           DIMENSION NOA(100),IC1(100),NAM1(100),NOA1(100),LAMDA1(100),
706          !TAU1(100)
707             COMMON /CB1/ A(1000),B(1000),PTA(15,4),PTB(101)
708           COMMON /CB2/ LENGA,MAXORD,NBE,NCS,NIE,NIFDT,NRE
709           COMMON /CB3/ C(1000),D(1000),F(1000),NAM(100)
710           COMMON /CB4/ QBE(100),QCS(100),PROBT,DELG(100),DF(100)
711           COMMON /CB5/ LAMDA(100),TAU(100)
712           INTEGER A,B,D,E,F,PTB,PTA
713           REAL LAMDA,LAMDA1
714           DO 1 J=1,NBE
715           NOA(J)=0
716           IC1(J)=0
717       1   CONTINUE
718           DO 2 J=1,LENGA
719           K=A(J)
720           NOA(K)=NOA(K)+1
721       2   CONTINUE
722           INBE=NBE-1
723           NRE=NBE
724           DO 3 I=1,INBE
725           IF(NOA(I).EQ.0)GO TO 4
726           GO TO 3
727       4   J=I + 1
728           NRE=NRE-1
729           DO 5 K=J,NBE
730           IC1(K)=IC1(K)+1
731       5   CONTINUE
732       3   CONTINUE
733           IF(NOA(NBE).EQ.0)NRE=NRE-1
734           DO 6 I=1,LENGA
735           J=A(I)
736           A(I)= A(I) - IC1(J)
737       6   CONTINUE
738           IF(NRE.EQ.NBE)GO TO 69
739           DO 7 I=1,NBE
740           IF(NOA(I).GT.0)GO TO 7
741           JRELV=I - IC1(I)
742           NOA(JRELV)=NOA(I)
743           LAMDA(JRELV) = LAMDA(I)
744           TAU(JRELV) = TAU(I)
745           NAM(JRELV) = NAM(I)
746       7   CONTINUE
747      69   CONTINUE
748           NIFDT=0
749           NFFDT=0
750           NIE=NRE
751           DO 70 I=1,NRE
752           IF(LAMDA(I).GT.0..AND.TAU(I).EQ.0)GO TO 71
753             IF(LAMDA(I).GT.0..AND.TAU(I).GT.0)GO TO 72
754           NIE=NIE-1
755           GO TO 70
756      71   CONTINUE
757           NIFDT=NIFDT+1
758           GO TO 70
759      72   CONTINUE
760           NFFDT=NFFDT+1
761      70   CONTINUE
762           IX=0
763           IY=NIFDT
764           IZ=NIE
765           DO 79 I=1,NRE
766           IF(LAMDA(I).GT.0..AND.TAU(I).EQ.0)GO TO 80
767           IF(LAMDA(I).GT.0..AND.TAU(I).GT.0.)GO TO 73
768           IZ=IZ+1
769           IC=IZ
770           GO TO 74
771      80     CONTINUE
772           IX=IX+1
773           IC=IX
774           GO TO 74
775      73   CONTINUE
776           IY=IY+1
777           IC=IY
778      74   IC1(I)=IC
779             NAM1(IC)=NAM(I)
780             NOA1(IC)=NOA(I)
```

```
      LAMDA1(IC)=LAMDA(I)
      TAU1(IC)=TAU(I)
79    CONTINUE
      DO 75 I=1,NRE
      NAM(I)=NAM1(I)
      NOA(I)=NOA1(I)
      LAMDA(I)=LAMDA1(I)
      TAU(I)=TAU1(I)
75    CONTINUE
      DO 76 I=1,LENGA
      J=A(I)
      K=IC1(J)
      A(I)=K
76    CONTINUE
      IF(NIE.EQ.NRE)GO TO 78
      IZ=NIE+1
      DO 77 I=IZ,NRE
      QBE(I)=TAU(I)
      DF(I)=0.
77    CONTINUE
78    CONTINUE
C PTB ARRAY
      IDUM=0
      PTB(1)=0
      IUP=NRE+1
      DO 8 I=2,IUP
      PTB(I)=NOA(I-1)+IDUM
      IDUM=PTB(I)
8     CONTINUE
      DO 9 I=1,NRE
      IC1(I)=PTB(I)
9     CONTINUE
      K=1
      DO 10 I=1,LENGA
      IF(I.LE.PTA(K,2))GO TO 11
      K=K+1
      GO TO 12
11    ICSN = (I - PTA(K,1))/K  +PTA(K,4) - PTA(K,3) + 1
      K1 = A(I)
      IC1(K1) = IC1(K1) + 1
      K2 = IC1(K1)
      B(K2)=ICSN
12    CONTINUE
10    RETURN
      END
C ***********************************************************
      SUBROUTINE EARRAY
C FOR EACH BASIC EVENT I, EARRAY IDENTIFIES THE
C BASIC EVENTS THAT ARE CONTAINED IN THE SAME CUT SETS
C WITH BASIC EVENT I
      COMMON /CB1/ A(1000),B(1000),PTA(15,4),PTB(101)
      COMMON /CB7/ ID(1000),E(1000),PTE(101)
      COMMON /CB2/ LENGA,MAXORD,NBE,NCS,NIE,NIFDT,NRE
      INTEGER A,B,D,E,F,PTB,PTA,PTE
      DIMENSION IDUM(100)
      IC=0
      PTE(1)=0
      DO 9 I=1,NIE
      IDUM(I)=0
      PTE(I)=0
9     CONTINUE
      DO 7 J1=1,NIE
      I2=MAXORD
      J2=J1+1
      INDA=PTB(J2)-PTB(J1)
      J=PTB(J2)+1
      I1=B(J-1)
      IF(I1.LE.PTA(1,2))GO TO 8
      DO 1 I=1,INDA
      J=J-1
      I1=B(J)
20      CONTINUE
      IF(I1.GE.PTA(I2,4)-PTA(I2,3)+1) GO TO 3
      I2=I2-1
      GO TO 20
3     I3 = I1 - PTA(I2,4) + PTA(I2,3) - 1
      IPTA = PTA(I2,1) + I3*I2 -1
      DO 4 K=1,I2
      IPTA=IPTA + 1
      I4 = A(IPTA)
      IF(I4.GT.NIE)GO TO 4
      IDUM(I4)=1
4     CONTINUE
1     CONTINUE
      IDUM(J1)=0
      DO 5 L=1,NIE
      IF(IDUM(L).EQ.0)GO TO 5
      IC = IC + 1
      E(IC)=L
5     CONTINUE
```

```
      PTE(J2)=IC
      DO 6 M=1,NIE
      IOUM(M)=0
    6 CONTINUE
      GO TO 7
    8 CONTINUE
      IC=IC+1
      PTE(J2)=IC
      E(IC)=0
    7 CONTINUE
      RETURN
      END
C ***********************************************************
      SUBROUTINE BEDATA(T,INTGX)
C BASIC EVENT DATA COMPUTED FOR TIME T, IF INTX=1 RENEWAL
C DENSITY WILL BE CALCULATED
      COMMON /CB2/ LENGA,MAXORD,NBE,NCS,NIE,NIFDT,NRE
       COMMON /CB4/ QBE(100),QCS(100),PROBT,DELG(100),DF(100)
      COMMON /CB5/ LAMDA(100),TAU(100)
      COMMON /CB6/ NTPT,TIME(8),PTOP(8),IDATA,NINTP(8)
      REAL LAMDA
       DIMENSION XMU(100)
      IF(NIFDT.EQ.0) GO TO 2
      DO 1 I=1,NIFDT
      QBE(I)= 1. - EXP(-LAMDA(I)*T)
    1 CONTINUE
    2 CONTINUE
      K=NIFDT + 1
      IF(K.GT.NIE)GO TO 4
      DO 3 I=K,NIE
      Q1= TAU(I)/(TAU(I)+1./LAMDA(I))
      QBE(I) = Q1*(1.-EXP(-LAMDA(I)*T/Q1))
    3 CONTINUE
    4 CONTINUE
      IF(INTGX.EQ.0)GO TO 10
      IF(NIFDT.EQ.0)GO TO 6
      DO 5 I=1,NIFDT
      DF(I)=LAMDA(I)*EXP(-LAMDA(I)*T)
    5 CONTINUE
    6 CONTINUE
      K=NIFDT +1
      IF(K.GT.NIE)GO TO 8
      DO 9 I=K,NIE
      XMU(I)= 1./LAMDA(I)
      DF(I)   = 1./(TAU(I) + XMU(I))  *(TAU(I)/(TAU(I)*XMU(I) + XMU(I)**
     12))*EXP(-(LAMDA(I)+1./TAU(I))*T)
    9 CONTINUE
    8 CONTINUE
   10 CONTINUE
      RETURN
      END
C ***********************************************************
      SUBROUTINE PTOPX
C PTOPX CALCULATES THE PROBABILITY OF THE TOP EVENT
      COMMON /CB1/ A(1000),B(1000),PTA(15,4),PTB(101)
      COMMON /CB2/ LENGA,MAXORD,NBE,NCS,NIE,NIFDT,NRE
      COMMON /CB4/ QBE(100),QCS(100),PROBT,DEL(100),DF(100)
      INTEGER A,B,O,E,F,PTB,PTE,PTA
      REAL LAMDA
      O=0.
      Z=1.
      DO 1 I=1,MAXORD
      J= MAXORD + 1 - I
      M= PTA(J,1) - 1
      ICSN= PTA(J,4)-PTA(J,3)
      J1=PTA(J,3)
      DO 2 I1=1,J1
      ICSN=ICSN+1
      DO 3 L=1,J
      M= M+1
      I2 = A(M)
      Z = Z*QBE(I2)
    3 CONTINUE
      QCS(ICSN)=Z
      G=Z + (1.-Z)*G
      Z=1.
    2 CONTINUE
    1   CONTINUE
      PROBT=G
      RETURN
      END
C ***********************************************************
      SUBROUTINE BRNBAUM(IBE)
C CALCULATE BIRNBAUM'S MEASURE OF IMPORTANCE
C FOR BSIC EVENT NUMBER IBE
      COMMON /CB1/ A(1000),B(1000),PTA(15,4),PTB(101)
      COMMON /CB2/ LENGA,MAXORD,NBE,NCS,NIE,NIFDT,NRE
      COMMON /CB4/ QBE(100),QCS(100),PROBT,DELG(100),DF(100)
      INTEGER A,B,O,E,F,PTB,PTE,PTA
      REAL LAMDA
```

```
      G= 0.
      G1=0.
      I2 = MAXORD
      INDA= PTB(IBE+1) - PTB(IBE)
      J= PTB(IBE + 1) + 1
      DO 1 I=1,INDA
      J=J-1
      I1= B(J)
      Z=QCS(I1)
      Z1= Z/QBE(IBE)
      G= Z +(1.-Z)*G
      G1= Z1 +(1.-Z1)*G1
    1 CONTINUE
      PKNEI = (PROBT - G) /(1.-G)
      DELG(IBE) = (1.- PKNEI)*G1
      RETURN
      END
C *****************************************************
      SUBROUTINE TROBT
C FIND TIME ROOTS FOR OPTIONS 2 OR 3
C CORRESPONDING TO INPUT DATA
      COMMON /CB1/ A(1000),B(1000),PTA(15,4),PTB(101)
      COMMON /CB2/ LENDA,MAXORD,NBE,NCS,NIE,NIFDT,NRE
      COMMON /CB4/ QBE(100),QCS(100),PROBT,DELG(100),DF(100)
      COMMON /CB5/ LAMDA(100),TAU(100)
      COMMON /CB6/ NTPT,TIME(8),PTOP(8),IDATA,NINTP(8)
      INTEGER A,B,D,E,F,PTB,PTE,PTA
      REAL LAMDA
      DIMENSION GP(3)
      T=1
      DO 1 I=1,NTPT
    2 CONTINUE
      DELX= .001*T
      T=T-DELX
      DO 3 J=1,3
      CALL BEDATA(T,0)
      CALL PTOPX
      GP(J)=PROBT
      T=T+DELX
    3 CONTINUE
      GPRIME = (GP(3) - GP(1))/(2.*DELX)
      T = T - .5*DELX -(GP(2)-PTOP(I))/GPRIME
      CALL BEDATA(T,0)
      CALL PTOPX
      IF(ABS((PTOP(1)-PROBT)/PTOP(I)).LT.1.E-3)GO TO 4
      GO TO 2
    4 CONTINUE
      PTOP(I)=PROBT
      TIME(I)=T
    1 CONTINUE
      RETURN
      END
C *****************************************************
      SUBROUTINE BPBE(IT1)
C BPBE CALCULATES THE BARLOW-PROSCHAN MEASURE OF BASIC
C EVENT IMPORTANCE-- IT1 IS AN INDEX NO. FOR TIME
      COMMON /CB1/ A(1000),B(1000),PTA(15,4),PTB(101)
      COMMON /CB2/ LENDA,MAXORD,NBE,NCS,NIE,NIFDT,NRE
      COMMON /CB3/ C(1000),D(1000),F(1000),NAM(100)
      COMMON /CB4/ QBE(100),QCS(100),PROBT,DELG(100),DF(100)
      COMMON /CB5/ LAMDA(100),TAU(100)
      COMMON /CB6/ NTPT,TIME(8),PTOP(8),IDATA,NINTP(8)
      INTEGER A,B,D,E,F,PTB,PTE,PTA
      REAL LAMDA
      T1=0.
      IC=NIE*((IT1-1)
      T2=TIME(IT1)
      IF(IT1.EQ.1)GO TO 1
      T1=TIME(IT1-1)
    1 CONTINUE
      IUP=(IT1+1
      KC=IC
      DO 2 J=1,IUP
      IF(J.EQ.1)GO TO 2
      TX=TIME(I-1)
      CALL BEDATA(TX,1)
      CALL PTOPX
      DO 3 J=1,NIE
      KC=KC+1
      CALL BRNBAUM(J)
      C(KC)= C(KC) + DELG(J)*DF(J)
    3 CONTINUE
      KC=IC
    2 CONTINUE
      YY=NINTP(IT1)/2
      DELT= (T2-T1)/YY
      T=T1
      X=2.
      IUP=NINTP(IT1)/2 - 1
      DO 4 I3=1,2
```

```
      DO 6 I4=1,IUP
      T= T+DELT
      CALL BEDATA(T,1)
      CALL PTOPX
      KC= IC
      DO 7 I5=1,NIE
      KC= KC +1
      CALL BRNBAUM(I5)
      C(KC)= C(KC) + DELG(I5)*X*DF(I5)
    7 CONTINUE
    6 CONTINUE
      T= T1-.5*DELT
      IUP=IUP+1
      X=4.
    4 CONTINUE
      XX=IUP-1
      TDELT =(T2-T1)/(6.*XX)
      KC=IC
      DO 8 J=1,NIE
      KC=KC+1
      C(KC) = C(KC)*TDELT
    8 CONTINUE
      IF(IT1.EQ.1)GO TO 11
      ID=IC
      IX=IC-NIE
       DO 10 II=1,NIE
      ID=ID+1
      IX=IX+1
      C(ID)=C(ID)+C(IX)
   10 CONTINUE
   11 CONTINUE
      RETURN
      END
C *****************************************************
      SUBROUTINE PATSRT(X,IXX,IYY,I1,I2)
C PATSRT SORTS ARRAYS IN DECENDING ORDER
      DIMENSION A(1000),IA(1000),X(1000),IXX(1000),IYY(1000)
      I=0
      N=I2-I1+1
      DO 1 J=I1,I2
      I=I+1
      A(I)=X(J)
    1 IA(I)=IXX(J)
      I=1
  100 M=I+I-1
      I=I+1
      IF(N-I)500,500,100
  500 M=ISR(M,1)
C THE ABOVE STATEMENT EQUIVALENT TO M=M/2
      IF(M) 450,200
  450 K=N-M
      DO 300 J=1,K
      I=J
  400 IF(A(I+M).LE.A(I))GO TO 300
      B=A(I)
      A(I)=A(I+M)
      A(I+M)=B
      IB= IA(I)
      IA(I)=IA(I+M)
      IA(I+M)=IB
      I=I-M
      IF(I) 300,300,400
  300 CONTINUE
      GO TO 500
  200 CONTINUE
      I=0
      DO 2 J=I1,I2
      I=I+1
      IXX(J)=IA(I)
    2 CONTINUE
      NUM=1
      J=I1
      IYY(I1)=1
      ILW=I1+1
      DO 3 I=ILW,I2
      K1=IXX(I)+I1-1
      K2=IXX(J)+I1-1
      IF(X(K1).EQ.X(K2))GO TO 4
      NUM=NUM+1
      J=I
      IYY(I)=NUM
      GO TO 3
    4 CONTINUE
      IYY(I)=IYY.J)
    3  CONTINUE
      RETURN
      END
C *****************************************************
      SUBROUTINE CONTRIB(IT1)
```

```
1141 C CONTRIB FINDS THE SEQUENTIAL CONTRIBUTORY
1142 C IMPORTANCE FOR BASIC EVENTS, IT1 IS AN INDEX
1143 C NUMBER FOR TIME
1144       COMMON /CB1/ A(1000),B(1000),PTA(15,4),PTB(101)
1145       COMMON /CB2/ LENGA,MAXORD,NBE,NCS,NIE,NIFDI,NRE
1146       COMMON /CB3/ C(1000),D(1000),F(1000),NAM(100)
1147       COMMON /CB4/ QBE(100),QCS(100),PROBT,DELG(100),DF(100)
1148       COMMON /CB5/ LAMDA(100),TAU(100)
1149       COMMON /CB6/ NTPT,TIME(8),PTOP(8),IDATA,NINTP(8)
1150       COMMON /CB7/ ID(1000),E(1000),PTE(101)
1151       INTEGER A,B,D,E,F,PTB,PTE,PTA
1152       REAL LAMDA
1153       T1=0.
1154       IC=NIE*(IT1-1)
1155       T2=TIME(IT1)
1156       IF(IT1.EQ.1)GO TO 1
1157       T1=TIME(IT1-1)
1158  1    CONTINUE
1159       IUP=IT1+1
1160       KC=IC
1161       DO 2 I=IT1,IUP
1162       IF(I.EQ.1)GO TO 2
1163       TX=TIME(I-1)
1164       CALL BEDATA(TX,1)
1165       DO 3 J=1,NIE
1166       KC=KC+1
1167       IY=PTE(J+1)
1168       IF(E(IY).EQ.0)GO TO 3
1169       IX=PTE(J)+1
1170       QBEI=QBE(J)
1171       QBE(J)=1.
1172       CALL PTOPX
1173       DO 30 K=IX,IY
1174       IZ=E(K)
1175       CALL BRNBAUM(IZ)
1176       C(KC) = C(KC) + DELG(IZ)*DF(IZ)*QBEI
1177  30   CONTINUE
1178       QBE(J)=QBEI
1179  3    CONTINUE
1180  2      CONTINUE
1181       YY=NINTP(IT1)/2
1182       DELT=(T2-T1)/YY
1183       T=T1
1184       X=2.
1185       IUP=NINTP(IT1)/2 - 1
1186       DO 4 I3=1,2
1187       DO 6 I4=1,IUP
1188       T=T+DELT
1189       CALL BEDATA(T,1)
1190       KC=IC
1191       DO 7 J=1,NIE
1192       KC=KC+1
1193       IY=PTE(J+1)
1194       IF(E(IY).EQ.0)GO TO 7
1195       IX=PTE(J)+1
1196       QBEI=QBE(J)
1197       QBE(J)=1.
1198       CALL PTOPX
1199       DO 70 K=IX,IY
1200       IZ=E(K)
1201       CALL BRNBAUM(IZ)
1202       C(KC)=C(KC)+DELG(IZ)*DF(IZ)*X*QBEI
1203  70   CONTINUE
1204       QBE(J)=QBEI
1205  7    CONTINUE
1206  6    CONTINUE
1207       T= T1+.5*DELT
1208       IUP=IUP+1
1209       X=4.
1210  4    CONTINUE
1211       XX=IUP-1
1212       TDELT=(T2-T1)/(6.*XX)
1213       KC=IC
1214       DO 8 J=1,NIE
1215       KC=KC+1
1216       C(KC)=C(KC)*TDELT
1217  8    CONTINUE
1218         IF(IT1.EQ.1) GO TO 11
1219       ID=IC
1220       IX=IC-NIE
1221       DO 10 I=1,NIE
1222       ID=ID+1
1223         IX=IX+1
1224       C(ID)=C(ID)+C(IX)
1225  10     CONTINUE
1226  11   CONTINUE
1227       RETURN
1228       END
```

```
1229 C ***********************************************
1230       SUBROUTINE BPCS(IBPMX,IT1,IBPNCS)
1231 C BPCS COMPUTES THE B-P MEASURE OF CUT SET IMPORTANCE
1232 C FOR CUT SETS OF ORDER ONE THROUGH IBPMX, IT1 IS AN
1233 C INDEX NUMBER FOR TIME
1234       COMMON /CB1/ A(1000),B(1000),PTA(15,4),PTB(100)
1235       COMMON /CB2/ LENGA,MAXORD,NBE,NCS,NIE,NIFDT,NRE
1236       COMMON /CB3/ C(1000),D(1000),F(1000),NAM(100)
1237       COMMON /CB4/ QBE(100),QCS(100),PROBT,DELG(100),DF(100)
1238       COMMON /CB5/ LAMDA(100),TAU(100)
1239       COMMON /CB6/ NTPT,TIME(8),PTOP(8),IDATA,NINTP(8)
1240       COMMON /CB7/ ID(1000),E(1000),PTE(101)
1241       DIMENSION QBEI(100),QCSI(100)
1242        REAL LAMDA
1243        INTEGER A,B,D,E,F,PTB,PTA,PTE
1244        T1=0.
1245        J=IBPMX+1
1246       DO 100 I=1,IBPMX
1247       J=J-1
1248       IF(PTA(J,1).NE.0)GO TO 101
1249 100   CONTINUE
1250       GO TO 10
1251 101   ICSBP=J
1252       IBPNCS =PTA(J,4)
1253       IC=IBPNCS+(IT1-1)
1254       T2=TIME(IT1)
1255       IF(IT1.EQ.1)GO TO 1
1256       T1=TIME(IT1-1)
1257 1     CONTINUE
1258       IUP =IT1 +1
1259       KC =IC
1260       ICSN = 0
1261       IA=0
1262       DO 2 I=IT1,IUP
1263       IF (I.EQ.1) GO TO 2
1264       TX=TIME(I-1)
1265        CALL BEDATA(TX,1)
1266       CALL PTOPX
1267       DO 11 II=1,ICSBP
1268       IF (PTA(II,1).EQ.0)GO TO 11
1269       ICSUP = PTA(II,3)
1270       DO 12 J=1,ICSUP
1271       ICSN = ICSN + 1
1272       IF(QCS(ICSN).EQ.0.) GO TO 12
1273       QCSI = 1
1274       KC = KC + 1
1275       DO 13 K=1,II
1276       IA = IA + 1
1277        L=A(IA)
1278       QBEI(L)=QBE(L)
1279       QCSI=QCSI*QBE(L)
1280       QBE(L)=1.
1281 13     CONTINUE
1282        IA=IA-II
1283       DO 14 K=1,II
1284       IA=IA+1
1285       L=A(IA)
1286       IF(L.GT.NIE)GO TO 14
1287       QBE(L)=QBEI(L)
1288       CALL PTOPX
1289       CALL BRNBAUM(L)
1290       C(KC) = C(KC) + (DELG(L))*(QCSI/QBEI(L))*DF(L)
1291       QBE(L)=1.
1292 14    CONTINUE
1293       IA = IA -II
1294       DO 15 K = 1,II
1295       IA=IA+1
1296       L=A(IA)
1297       QBE(L)=QBEI(L)
1298 15    CONTINUE
1299       QCS(ICSN) = QCSI
1300 12     CONTINUE
1301 11    CONTINUE
1302       T1=TIME(IT1 - 1)
1303       KC = IC
1304       ICSN=0
1305        IA=0
1306 2     CONTINUE
1307       YY=NINTP(IT1)/2
1308        DELT=(T2-T1)/YY
1309       T = T1
1310        X=2.
1311       IUP =NINTP(IT1)/2 - 1
1312       DO 4 I3=1,2
1313       DO 6 I4= 1,IUP
1314       T=T+DELT
1315       KC = IC
1316       ICSN=0
1317        IA=0
1318        CALL BEDATA(T,1)
1319       DO 21 II=1,ICSBP
1320       IF (PTA(II,1).EQ.0)GO TO 21
```

```
      ICSUP = PTA(I1,3)
      DO 22 J=1,ICSUP
      ICSN = ICSN + 1
      IF(QCS(ICSN).EQ.0.) GO TO 22
      QCSI= 1.
      KC = KC + 1
      DO 23 K=1,I1
      IA = IA + 1
       L=A(IA)
      QBEI(L)=QBE(L)
      QCSI=QCSI*QBE(L)
      QBE(L)=1.
23    CONTINUE
       IA = IA - I1
      DO 24 K=1,I1
      IA=IA+1
      L=A(IA)
      IF(L.GT.NIE)GO TO 24
      QBE(L)=QBEI(L)
      CALL PTOPX
      CALL BRNBAUM(L)
      C(KC) = C(KC) + (DELG(L))*(QCSI/QBEI(L))*DF(L)*X
      QBE(L)=1.
24    CONTINUE
      IA = IA -I1
      DO 25 K = 1,I1
      IA=IA+1
      L=A(IA)
      QBE(L)=QBEI(L)
25    CONTINUE
      QCS(ICSN) = QCSI
22    CONTINUE
21     CONTINUE
6      CONTINUE
       T = T1 - .5*DELT
      IUP=IUP+1
       X=4.
4      CONTINUE
      XX=IUP-1
       TDELT= (T2-T1)/(6.*XX)
       KC = IC
      ICSN=0
       DO 8 J=1,NIE
       KC = KC+1
        C(KC) = C(KC)*TDELT
8      CONTINUE
       IF(IT1.EQ.1) GO TO 10
       ID = IC
      IX=IC-IBPNCS
        DO 9 I=1,IBPNCS
       ID = ID +1
      IX=IX-1
       C(ID)=C(IX) + C(ID)
9      CONTINUE
10     CONTINUE
       RETURN
       END
C ***********************************************************
      SUBROUTINE SVBE(IT1)
C SVBE COMPUTES THE RUSSELL-VESELY IMPORTANCE FOR BASIC
C EVENTS.  IT1 IS AN INDEX FOR TIME
      COMMON /CB1/ A(1000),B(1000),PTA(15,4),PTB(100)
      COMMON /CB2/ LENGA,MAXORD,NBE,NCS,NIE,NIFDT,NRE
      COMMON /CB3/ C(1000),D(1000),F(1000),NAM(100)
      COMMON /CB4/ QBE(100),QCS(100),PROBT,DELG(100),DF(100)
      COMMON /CB5/ LAMDA(100),TAU(100)
      COMMON /CB6/ NTPT,TIME(8),PTOP(8),IDATA,NINTP(8)
      INTEGER A,B,D,E,F,PTB,PTE,PTA
      REAL LAMDA
      T=TIME(IT1)
      I2=MAXORD
      IBE=NIE*(IT1-1)
      CALL BEDATA(T,0)
      CALL PTOPX
      DO 1 I=1,NIE
      IBE=IBE+1
      INDA=PTB(I+1) - PTB(I)
      J=PTB(I+1) + 1
      G=0.
      DO 2 K=1,INDA
      J=J-1
      I1=B(J)
      Z=QCS(I1)
      G=Z+(1.-Z)*G
2      CONTINUE
      C(IBE)=G/PROBT
1      CONTINUE
      RETURN
      END
```

```
      SUBROUTINE POINTS
C POINTS FINDS THE NUMBER OF INTEGRATION POINTS FOR
C THE TIME INTEGRATED MEASURES OF IMPORTANCE
      COMMON /CB6/ NTPT,TIME(8),PTOP(8),IDATA,NINTP(8)
      PROB=1.E-6
      DO 1 I=1,NTPT
      NINTP(I)=10
      Y=PTOP(I)/PROB
      K=10.*ALOG10(Y)
      IF(K.GT.10)NINTP(I)=(K/2)*2
      IF(K.GT.100)NINTP(I)=100
      PROB=PTOP(I)
    1 CONTINUE
      RETURN
      END
C ***********************************************************
      SUBROUTINE SORT(ARRAY,N)
C REARRANGING DATA FOR PLOTTING OPTION
      DIMENSION ID(8),ARRAY(8),CX(8),IC(8)
      DO 3 I=1,N
      IC(I)=I
      ID(I)=I
    3 CONTINUE
      CALL PATSRT(ARRAY,IC,ID,1,N)
      DO 1 I=1,N
      CX(I)=ARRAY(I)
    1 CONTINUE
      J=N+1
      DO 2 I=1,N
      J=J-1
      JI=IC(J)
      ARRAY(I)=CX(JI)
    2 CONTINUE
      RETURN
      END
C ***********************************************************
      SUBROUTINE SWAPN
C REARRANGING DATA FOR PRINTOUT
      COMMON /CB3/ C(1000),D(1000),F(1000),NAM(100)
      COMMON /CB6/ NTPT,TIME(8),PTOP(8),IDATA,NINTP(8)
      COMMON /CB8/ INDEX,IPR,W,FACTOR,IPR
      DIMENSION CC(1000)
      INTEGER D
      DO 1 I=1,NTPT
      K=(I-1)*INDEX
      DO 2 L=1,INDEX
      K=K+1
      CC(L)=C(K)
    2 CONTINUE
      K=(I-1)*INDEX
      DO 3 L=1,INDEX
      K=K+1
      M=D(K)
      C(K)=CC(M)
    3 CONTINUE
    1 CONTINUE
      RETURN
      END
C ***********************************************************
      SUBROUTINE SSBP(SSRBD,IERR)
C SSBP COMPUTES THE STEADY STATE B-P MEASURE
C OF BASIC EVENT IMPORTANCE
      COMMON /CB3/ C(1000),D(1000),F(1000),NAM(100)
      COMMON /CB2/ LENGA,MAXORD,NBE,NCS,NIE,NIFDT,NRE
      COMMON /CB4/ OSE(100),OCS(100),PROBT,DELG(100),DF(100)
      COMMON /CB5/ LAMDA(100),TAU(100)
      DIMENSION XMU(100)
      INTEGER D,F
      REAL LAMDA
      IF(NIFDT.EQ.0)GO TO 3
      DO 2 I=1,NIFDT
      QBE(I)=1.
      C(I)=0.
      D(I)=I
    2 CONTINUE
      CALL PTOPX
      IF(PROBT.LT .999999)GO TO 3
      IERR=1
      RETURN
    3 CONTINUE
      ILW=NIFDT+1
      DO 4 I=ILW,NIE
      D(I)=I
      XMU(I)=1./LAMDA(I)
      QBE(I)=TAU(I)/(XMU(I)+TAU(I))
    4 CONTINUE
      CALL PTOPX
      SSRBD=0.
      DO 5 I=ILW,NIE
      CALL BRNBAUM(I)
```

```
      C(I)=DELG(I)/(XMU(I)+TAU(I))
      SSRBD=C(I)+SSRBD
    5 CONTINUE
      DO 6 I=ILW,NIE
      C(I)=C(I)/SSRBD
    6 CONTINUE
      CALL PATSRT(C,D,F,1,NIE)
      CALL SWAPN
      RETURN
      END
C     ***************************************************
      SUBROUTINE PLOTS
      COMMON /CB3/ C(1000),D(1000),F(1000),NAM(100)
      COMMON /CB6/ NTPT,TIME(8),PTOP(8),IDATA,NINTP(8)
      COMMON /CB7/ ID(1000),E(1000),PTE(101)
      COMMON /CB8/ INDEX,IPRLW,FACTOR,IPR
      DIMENSION KPLOT(1000)
      INTEGER D
      DO 5 I=1,INDEX
      KPLOT(I)=0
      ID(I)=0
    5 CONTINUE
      DO 1 I=1,NTPT
      J=(I-1)*INDEX+1
      J1=D(J)
      J2=J+J1-1
      XMAX=C(J2)
      ILOW=J
      IUP=I*INDEX
      DO 2 K=ILOW,IUP
      J1=D(K)
      J2=ILOW+J1-1
      XREF=C(J2)
      XQ=XREF/XMAX
      IF(XQ.LT.FACTOR)GO TO 1
      IREF=D(K)
      KPLOT(IREF)=1
    2 CONTINUE
    1 CONTINUE
      IPR=D
      DO 4 I=1,INDEX
      IF(KPLOT(I).NE.1)GO TO 4
      IPR=IPR+1
      ID(IPR)=I
    4 CONTINUE
      RETURN
      END
```

REFERENCES Appendix A

[A-1] W. E. Vesely and R. E. Narum, *PREP and KITT: Computer Codes for the Automatic Evaluation of Fault Trees*, Idaho Nuclear Corp., Idaho Falls, Rept. IN 1349 (1970).

[A-2] J. Murchland, "Fundamental Probability Relations for Repairable Items," *NATO Advanced Study Institue on Generic Techniques in Systems Reliability Assessment*, The University of Liverpool, July 17-27, 1973.

APPENDIX B

EXAMPLE OF SYSTEM UPGRADE

We chose the well known pressure tank example due to Haasl[B-1] for purposes of system upgrade. The description and schematic of the system is given in Fig. B.1. A hazard associated with the operation of this system is a pressure tank rupture. Internal overpressure sufficient to rupture the tank occurs if the pump runs for a period greater than 60 seconds. Fig. B.2 shows a fault tree that identifies all the basic causes leading to a tank rupture. We limit our discussion in considering primary events (i.e., circles) or hardware failures that are numbered one through six on the fault tree in Fig. B.2. A reduced version of this fault tree is shown in Fig. B.3. The path sets and cut sets are identified. The min path set respresentation is given by

$$g_0(F(t)) = \left[\coprod_{i=1,2,3} F_i(t)\right]\left[\coprod_{i=1,2,4,5,6} F_i(t)\right].$$

The corresponding proportional hazards for the basic events are shown in Table B-1. We see that there are two events that are single even. cut sets, event 1, "pressure tank ruptures under load" and event 2, "K2 relay contacts fail to open." There are no design changes in the system that can eliminate event 1 being a single order cut set.* Rigid quality control and periodic inspection of the pressure tank could slightly reduce the probability of this event. More important, however, is event 2 that is 1000 times more likely to occur than event number 1. For each primary event in the original fault tree we plot $\frac{\alpha_i}{g_0(\underline{\alpha}, q(t))} \cdot \frac{\partial g_0(\underline{\alpha}, q(t))}{\partial \alpha_i}$

*Event 1 is an inherent failure of a system element exercised within its design envelope.

TABLE B-1

PROPORTIONAL HAZARDS FOR PRESSURE TANK FAULT TREES

Event	Event No.	Proportional Hazards (α_i)
Pressure Tank Ruptures Under Load	1	.001
K2 Relay Contact Fails to Open	2	1
Pressure Switch Contacts Fail to Open	3	1
Timer Contacts Fail to Open	4	1
K1 Relay Contacts Fail to Open	5	1
S1 Switch Contacts Fail to Open	6	1
Pressure Relief Valve Jammed Closed	7	10

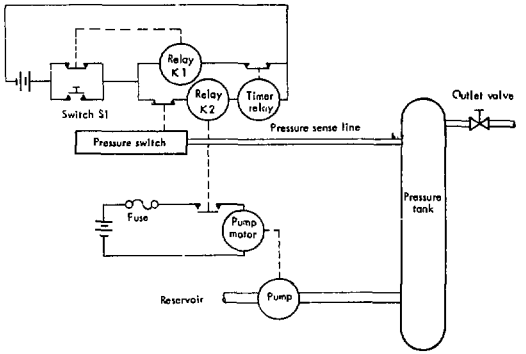
versus $g_0(\underline{\alpha}, q(t))$. We see according to Fig. B.4, that event 2 is always more important than any other of the primary events (a result that is expected solely on the basis of the visual inspection of the fault tree). To reduce the criticality of event 2, we propose two alternate designs for the pressure tank system, design X and design Y. In design X, to compensate for the failure of the K2 contacts, we install a relief valve on the pressure tank. We see in Fig. B.6 that the order of each cut set increases by one (except {1}).

If we want a more reliable design than design X, fig. B.5 tells us that the system is optimally upgraded by reducing the importance of the K2 relay failure. For reliable designs we see that the relief valve failure is of equal importance as the K2 relay failure. In practice,

however, we cannot install a more reliable relief valve as indicated in Table B-1. In design Y, as shown in Fig. B-7, we modify the control circuit so that the failure of the K2 contacts by itself is not catastrophic. Also in design Y we install a relief valve on the pressure tank. The fault tree for design Y is given in Fig. B.8.

Fig. B.9 shows that the system is improved with either design X or Y. Design Y is more reliable than design X. Design Y can be operated longer that design X before system degradation occurs.

We see that the assumption of proportional hazards permits a more powerful form of decision making than qualitative judgments based on the inspection of the minimal cut sets. It is more evident to management by inspection of the plots of the upgrading function (see Fig. B.4 and B.5) where weaknesses in the system exist.



Original Pressure Tank System Design

Fig. B.1 Description and Schematic of Original Pressure Tank System

The system is designed to make hydraulic energy available from the tank for some external load at some specified range of pressures whenever the reset switch is closed. The system performs two functions; a pumping function and a monitoring function. When the

reset switch S1 is momentarily closed, the coil of the power relay No. 1 is energized and the relay contacts are latched closed, providing continuous power to the monitoring circuit. Simultaneously, the coils of power relay No. 2 and the timer are energized; the contacts of power relay No. 2 are then closed, power is supplied to the pump motor, and pumping is initiated. At the same time, the timing cycle is initiated. When the tank pressure reaches some specified value, the contacts of the pressure-sensing switch open, de-energizing the coil of power relay No. 2; this causes the contacts to the pumping circuit to open, and pumping stops. At the same instant, the timer coil is de-energized, and the timer resets to zero. When the tank pressure drops below some specified lower pressure, the contacts of the pressure sensing switch close, power relay No. 2 and the timer are re-energized, and the pumping cycle is reinitiated. If for some reason the pressure-sensing switch fails to open, the timing cycle will run out, opening the circuit to power relay No. 1; its contacts are then unlatched and opened, and current is denied to power relay No. 2. Again, pumping ceases. The pumping can only be initiated by closing the reset switch.

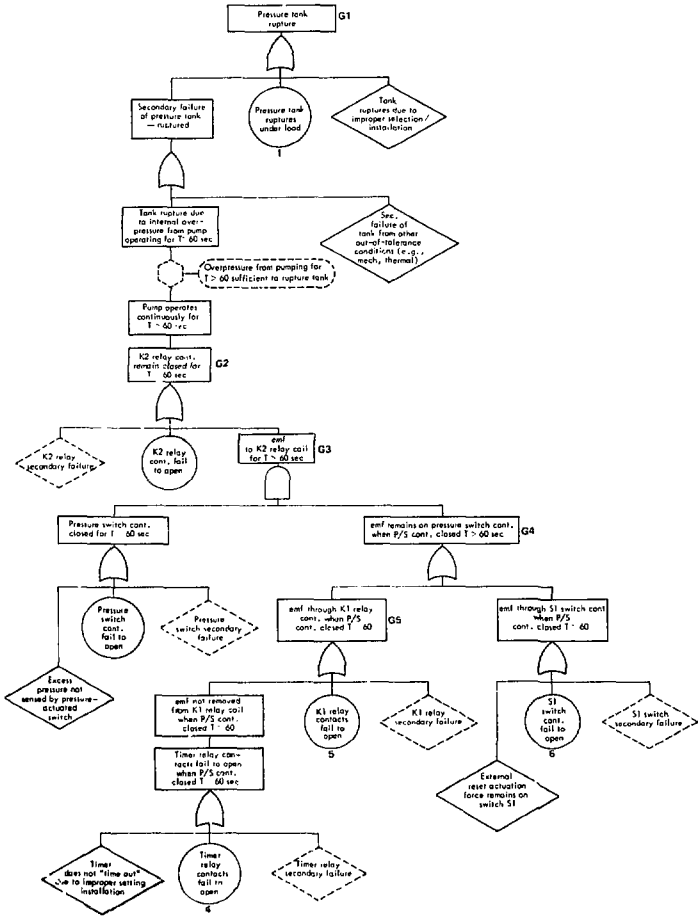


Fig. B.2 Fault Tree for Original Pressure Tank System

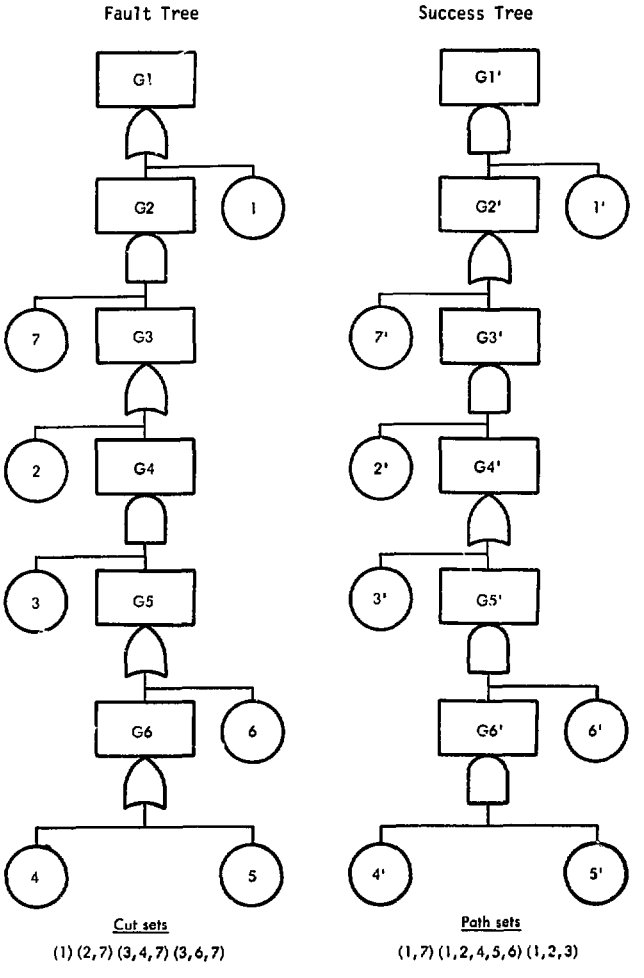


Fig. 8.3 Reduced Version of Original Fault Tree

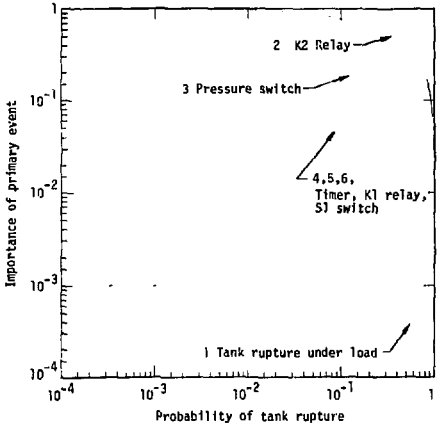


Fig. B.4 Plots of Upgrading Function for for Original Pressure Tank Design

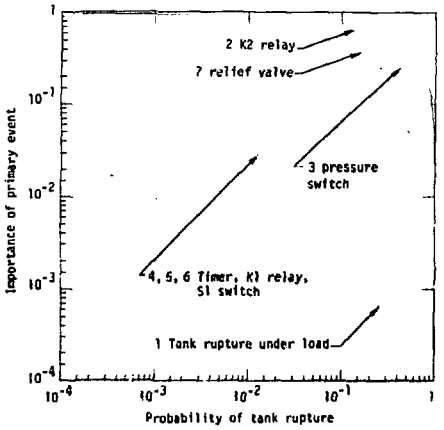


Fig. B.5 Plots of Upgrading Function for Design X of Pressure Tank

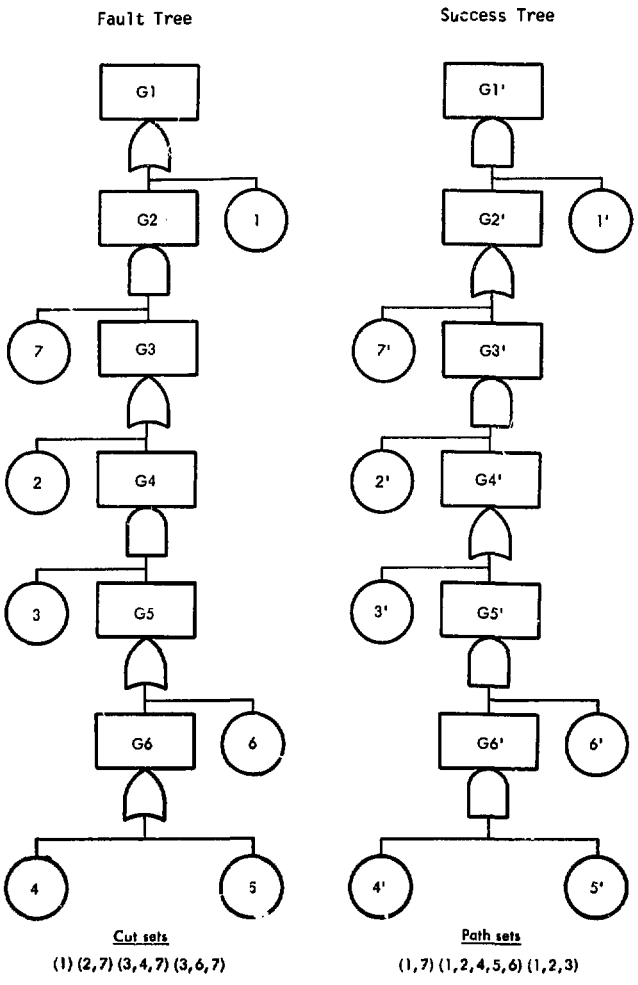


Fig. B.6 Reduced Version of System X Fault Tree

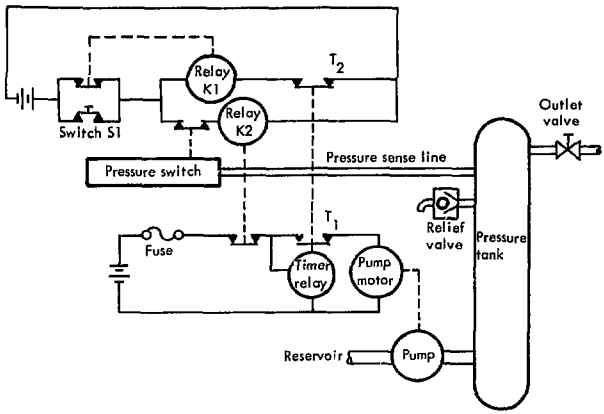


Fig. B.7 Description and Schematic of Design Y

Modified Design Y

The modified design works on the same principle as the original design with the exception of the timer relay. The timer relay is designed to time out 60 seconds after the pumping cycle is initiated. If current is interrupted to the timer relay before the timer times out, the timer relay resetsitself and the T1 and T2 contacts remain closed. If the timer times out, the contacts T1 and T2 open. If current is interrupted to the timer when the contacts T1 and T2 open, then a limiting device inside the timer closes the T1 and T2 contacts and resets the timer. If current is not interrupted to the timer when the T1 and T2 contacts open, then the timer relay must be manually reset. Table B-2 is given to assist the reader in understanding the pressure tank operation of the modified design.

TABLE B-2 Control Circuit Description Design Y

Mode of System Operation	Position of K2 Contacts	Position of T1 and T2 Contacts	Action required to reinitiate pumping cycle
Normal Operation	Opens when pressure switch opens	Remains closed	no action (automatic
Pressure Switch Contacts Fail to open	Open when contacts T2 open	Open momentarily when timer times out	Press reset switch S1
K2 relay contacts fail to open	Closed when timer times out	Open when timer times out and closes when timer relay is reset manually	Manually reset timer relay

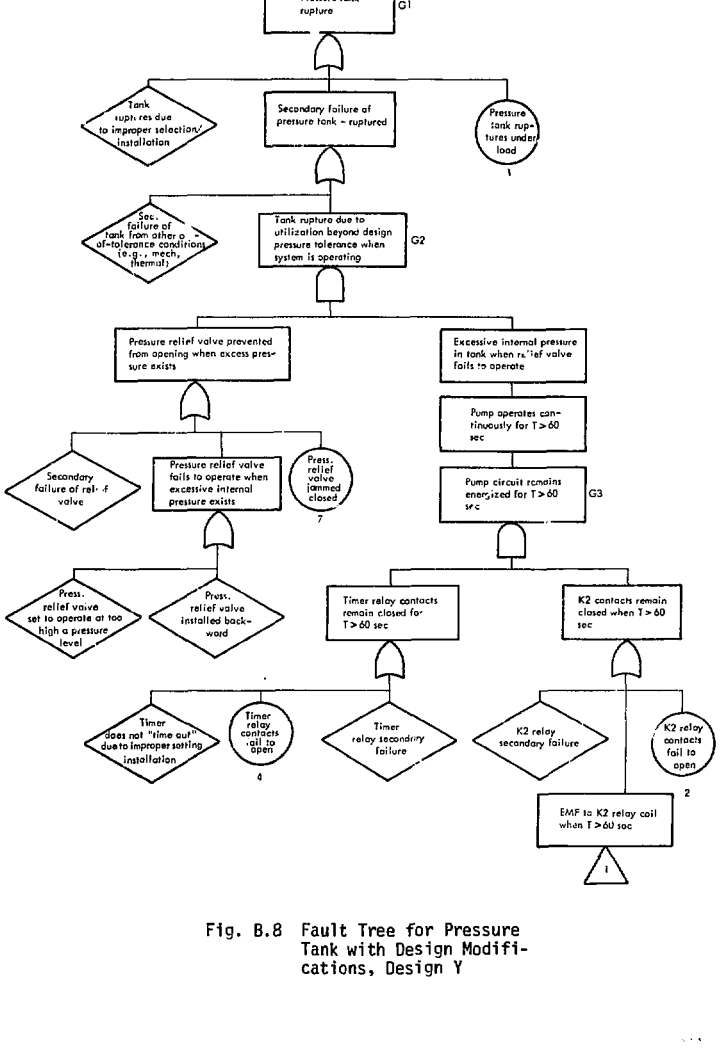


Fig. B.8 Fault Tree for Pressure Tank with Design Modifications, Design Y

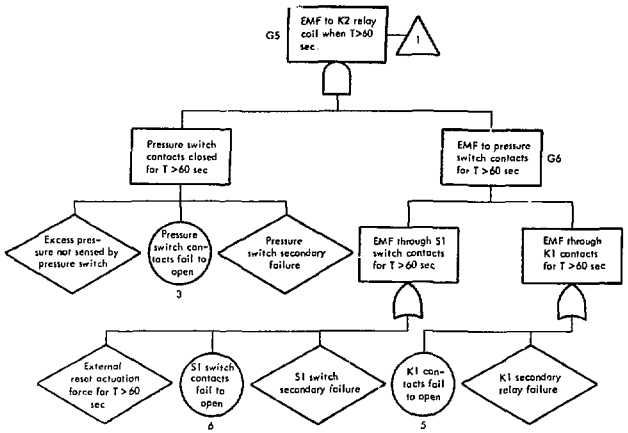

G5
EMF to K2 relay coil when T>60 sec.
1
Pressure switch contacts closed for T >60 sec
EMF to pressure switch contacts for T >60 sec
G6
Excess pressure not sensed by pressure switch
Pressure switch contacts fail to open
3
Pressure switch secondary failure
EMF through S1 switch contacts for T >60 sec
EMF through K1 contacts for T >60 sec
External reset actuation force for T >60 sec
S1 switch contacts fail to open
6
S1 switch secondary failure
K1 contacts fail to open
5
K1 secondary relay failure

Fig. B.8 Cont'd

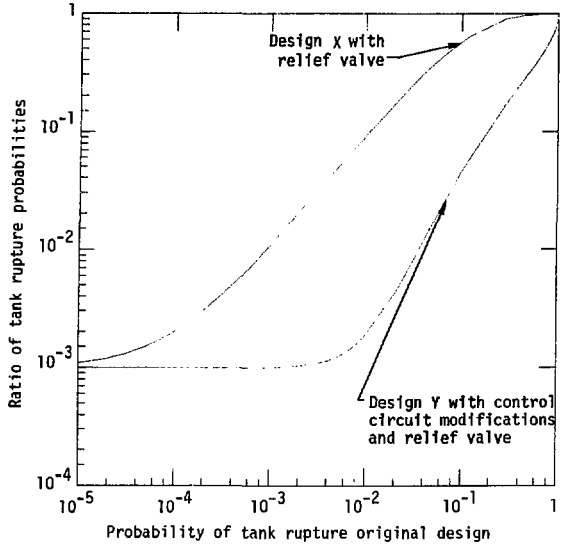


Fig. B.9 g_X/g_o vs. g_o; g_Y/g_o vs. g_o

where g_o is the probability of tank rupture, original design

g_X is the probability of tank rupture, design X

g_Y is the probability of tank rupture, design Y

REFERENCES Appendix B

[B-1] D. F. Haasl, Institute of Systems Sciences, Bellevue, Wash., private communication (1972).

APPENDIX C

OPTIMAL SENSOR LOCATION FOR TRIGA SCRAM CIRCUIT

We briefly describe the operation of the TRIGA reactor and the scram control circuit [C-1]. A fault tree is given with top event "Failure to Scram." The sequential contributory importance of each basic event in the fault tree is computed to show the optimal location of preventive sensors in the scram circuit.

C.1 TRIGA Nuclear Reactor

The UC-B TRIGA reactor used as an example here is a swimming pool-type reactor located in the basement of Etcheverry Hall on the Berkeley campus and is operated by the Department of Nuclear Engineering. The reactor can operate at power levels as high as one megawatt at steady state and can be pulsed to 1,200 megawatts.

C.2 Scram Circuit

A simplified diagram of the TRIGA scram circuit is shown in Fig. C.1. The circuit delivers current to the control magnets and solenoid valve of the transient rod.

The operator pushes the "power on" switch that energizes relay coil R-16, closing relay contacts K16A and K16B. When the operate key switch is placed in the reset position, it momentarily energizes relays R19 and R20, which, in turn, energizes relays R7 to R12. By spring action the reset switch returns to the "on" position. The lower "B" contacts of each of the relays receives voltage from one of the corresponding instrument channels and will apply this voltage to their coils, thus maintaining the coils energized. The upper "A" contacts will establish the relay K7 circuit which provides power to the magnets and solenoid valve. When any

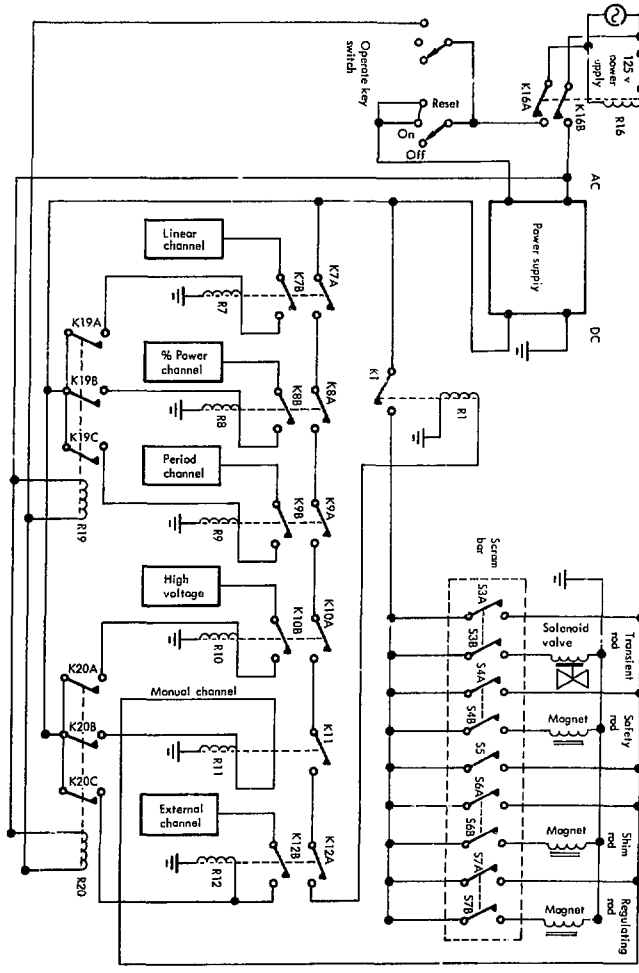


Fig. C.1 TRIGA Scram Circuit

instrumentation channel interrupts its voltage supply to the corresponding relay, a scram should occur, i.e., if any of the scram magnets or the solenoid valve are de-energized, then their respective control rods should drop into core.

C.3 TRIGA Fault Tree

Shown in Fig. C.2 is a fault tree that describes the possible combination of events that can cause the reactor not to scram when the maximum permissible power level of one megawatt at steady state is exceeded. Failure to scram means failure to insert an adequate number of control rods in the core to effectively shut down the nuclear reaction. In the case of the TRIGA, at least two of the four control rods must be successfully inserted for successful shutdown. Three of the four control rods drop into the core when their respective scram magnets are de-energized. The fourth control drops when its air chamber is depressurized by de-energizing a solenoid valve. The three instrument channels capable of de-energizing the scram magnets and solenoid valve are the linear channel, the per cent power channel and the period channel (if power increases at a rate faster than a factor of e in three seconds). The fault tree as shown does not allow for operator intervention.

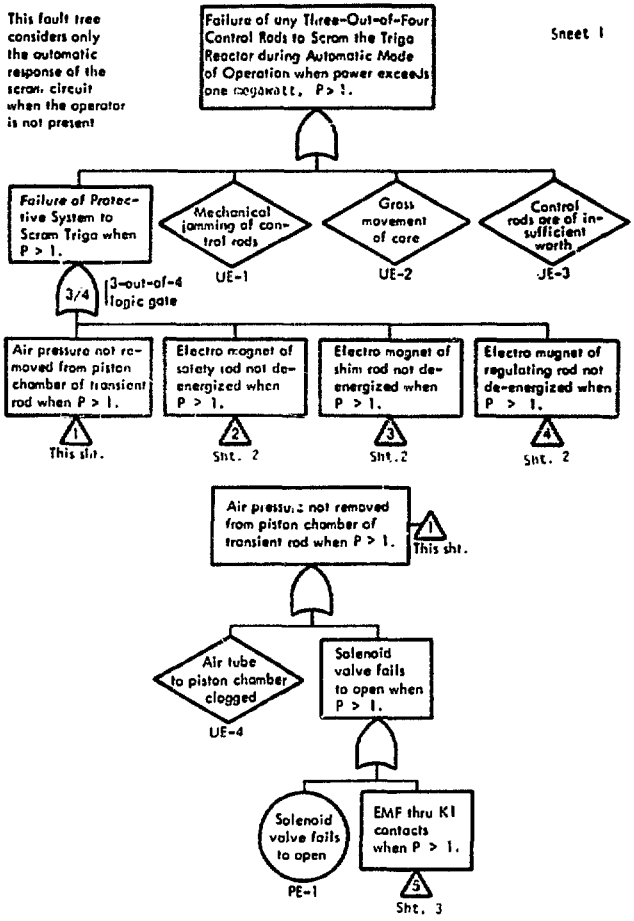


Fig. C.2 TRIGA Fault Tree

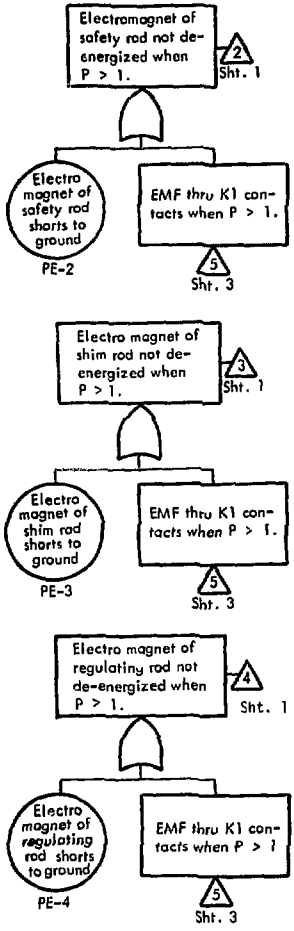

Sheet 2
Electromagnet of safety rod not de-energized when P > 1.
2
Sht. 1
Electro magnet of safety rod shorts to ground
PE-2
EMF thru K1 contacts when P > 1.
5
Sht. 3
Electro magnet of shim rod not de-energized when P > 1.
3
Sht. 1
Electro magnet of shim rod shorts to ground
PE-3
EMF thru K1 contacts when P > 1.
5
Sht. 3
Electro magnet of regulating rod not de-energized when P > 1.
4
Sht. 1
Electro magnet of regulating rod shorts to ground
PE-4
EMF thru K1 contacts when P > 1
5
Sht. 3

Fig. C.2 Cont'd.

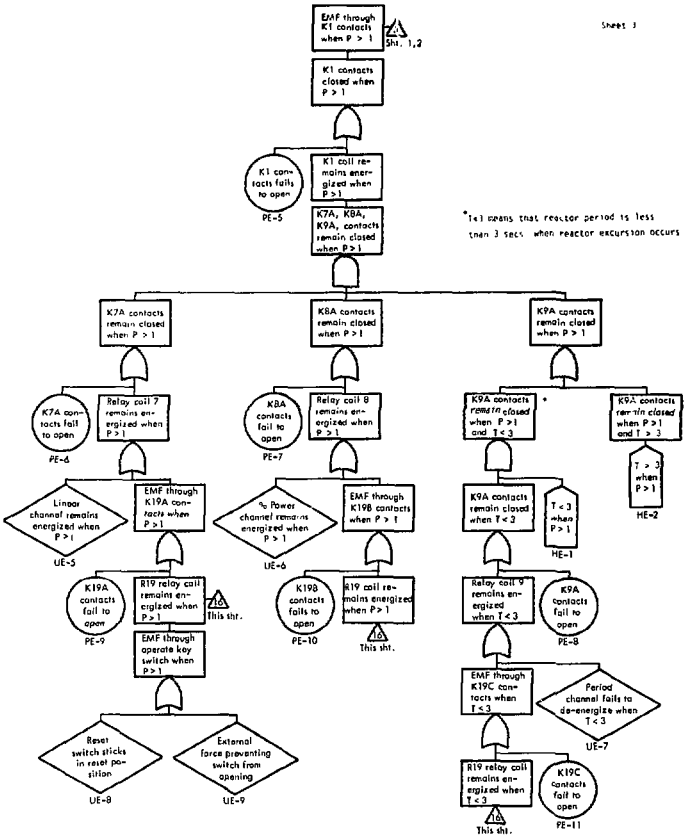

EMF through K1 contacts when P > 1
Sht. 1,2
Sheet 3
K1 contacts closed when P > 1
K1 contacts fails to open
PE-5
K1 coil remains energized when P >1
K7A, K8A, K9A, contacts remain closed when P>1
*T<1 means that reactor period is less than 3 secs. when reactor excursion occurs
K7A contacts remain closed when P > 1
K8A contacts remain closed when P > 1
K9A contacts remain closed when P > 1
K7A contacts fail to open
PE-6
Relay coil 7 remains energized when P > 1
K8A contacts fail to open
PE-7
Relay coil 8 remains energized when P > 1
K9A contacts remain closed when P >1 and T < 3
K9A contacts remain closed when P >1 and T > 3
T > 3 when P > 1
HE-2
Linear channel remains energized when P > 1
UE-5
EMF through K19A contacts when P >1
% Power channel remains energized when P > 1
UE-6
EMF through K19B contacts when P > 1
K9A contacts remain closed when T < 3
T < 3 when P > 1
HE-1
K19A contacts fail to open
PE-9
R19 relay coil remains energized when P >1
16 This sht.
K19B contacts fails to open
PE-10
R19 coil remains energized when P > 1
16 This sht.
Relay coil 9 remains energized when T < 3
K9A contacts fail to open
PE-8
EMF through operate key switch when P > 1
EMF through K19C contacts when T < 3
Period channel fails to de-energize when T < 3
UE-7
Reset switch sticks in reset position
UE-8
External force preventing switch from opening
UE-9
R19 relay coil remains energized when T < 3
16 This sht.
K19C contacts fail to open
PE-11

Fig. C.2 Cont'd

C.4 Input Data to the IMPORTANCE Computer Code

As shown in Fig. C.3, all basic events are assumed to have an infinite fault duration time (as indicated by the third column which is all zeros except for the house events). The failure rate data in the second column is expressed as failures per cycle (assuming 300 cycles/year). As seen by the listing of the minimal cut sets in Fig. C.3, there is a great deal of redundancy in the scram circuit. There is only one min cut set of order one involving failure of an active component. This cut set is primary event PE-5, failure of the K1 contacts to open.

C.5 Output of IMPORTANCE Code

Data points generated from the IMPORTANCE code are plotted in Fig. C.4. The sequential contributory importance versus operating cycles is plotted for the nine basic events with the highest ranking. It is shown that the linear power channel and per cent power channel are the greatest contributors to system failure. Of nearly equal importance is the period channel.

On the basis of the above results, the optimal locations in the TRIGA scram circuit for preventive sensors are the linear power and per cent power channels. Almost equal consideration should be given to the period channel.

```
      TRIGA FAULT TREE FOR AUTOMATIC SCRAM
            1                   6
1.000E+03 1.500E+03 2.000E+03 2.500E+03 3.000E+03 3.500E+03 4.000E+03 4.500E+03
            0         0         0         0         1         1
            0         0
            0         0
            1     .00010
           22        81
            1 1.0000E-05 .0000E+00   UE- 8
            2 1.0000E-05 .0000E+00   UE- 9
            3 1.0000E-05 .0000E+00   UE- 4
            4 1.0000E-04 .0000E+00   PE- 1
            5 1.0000E-05 .0000E+00   PE- 2
            6 1.0000E-05 .0000E+00   PE- 3
            7 1.0000E-05 .0000E+00   PE- 4
            8 1.0000E-05 .0000E+00   PE-11
            9 1.0000E-04 .0000E+00   UE- 7
           10 1.0000E-05 .0000E+00   PE- 8
           11 1.0000E-05 .0000E+00   PE-10
           12 1.0000E-05 .0000E+00   PE- 9
           13   .0   E+00 .5000E+00   HE- 1
           14 1.0000E-04 .0000E+00   UE- 6
           15 1.0000E-04 .0000E+00   UE- 5
           16   .0   E+00 .5000E+00   HE- 2
           17 1.0000E-05 .0000E+00   PE- 7
           18 1.0000E-05 .0000E+00   PE- 6
           19 1.0000E-05 .0000E+00   PE- 5
           20 1.0000E-06 .0000E+00   UE- 1
           21 1.0000E-06 .0000E+00   UE- 2
           22 1.0000E-06 .0000E+00   UE- 3
 19
 20
 21
 22
  1   13
  2   13
  2   16
  1   16
  5    6    7
 11   12   16
 11   15   16
 11   16   18
 12   14   16
 14   15   16
 14   16   18
 12   16   17
 15   16   17
 16   17   18
  3    6    7
  4    6    7
  3    5    7
  4    5    6
  4    5    6
 10   11   12   13
 10   11   13   15
 10   11   13   18
 10   12   13   14
 10   13   14   15
 10   13   14   18
 10   12   13   17
 10   13   15   17
 10   13   17   18
  9   11   12   13
  9   11   13   15
  9   11   13   18
  9   12   13   14
  9   13   14   15
  9   13   14   18
  9   12   13   17
  9   13   15   17
  9   13   17   18
  8   11   12   13
  8   11   13   15
  8   11   13   18
  8   12   13   14
  8   13   14   15
  8   13   14   18
  8   12   13   17
  8   13   15   17
  8   13   17   18
```

Basic Event Data

Fault Duration Time

Min Cut Sets

Fig. C.3 Input Data for Importance Computer Code

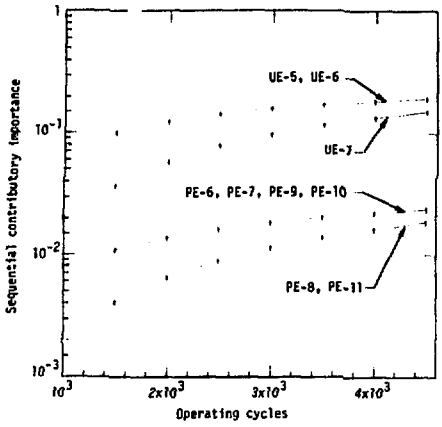


Fig. C.4 Plots of Contributory Importance for TRIGA Scram Circuit

Basic Event*	Description	Failure Rate (Per Cycle)
UE-5	Linear Channel Remains Energized when P > 1	10^{-4}
UE-6	% Power Channel Remains Energized when P > 1	10^{-4}
UE-7	Period Channel Fails to De-energize when T < 3 (Reactor Period Less Than Three Seconds)	10^{-4}
PE-6	K7A Contacts Fail to Open	10^{-5}
PE-7	K8A Contacts Fail to Open	10^{-5}
PE-9	K19A Contacts Fail to Open	10^{-5}
PE-10	K19B Contacts Fail to Open	10^{-5}
PE-8	K9A Contacts Fail to Open	10^{-5}
PE-11	K19C Contacts Fail to Open	10^{-5}

*All basic events listed appear on Sheet 3 of the Fig. C.2 Fault Tree

REFERENCES Appendix C

[C-1] Reactor Safety Analysis, Dept. of Nuclear Engineering, University of California, Berkeley, (1964).

APPENDIX D

DIAGNOSTIC SENSORS IN A CHEMICAL PROCESSING SYSTEM

In Appendix C we considered placing sensors on components in the scram circuit of the TRIGA reactor. We showed how system failure can be prevented by detecting failure of critical redundant components that have a tendency of failing prior to system failure. The occurrence of a min cut set in the TRIGA fault tree at the time of a scram demand implies that system failure is to occur instantaneously. We now consider systems that have a finite response time to system fault conditions before system failure occurs. In the chemical processing system given in Fig. D.1, we assume there is a finite response time between the occurrence of a min cut set and the occurrence of the top event.* In this system we are concerned that a chemical reactor explosion will occur as the result of an exothermic chemical reation. The fault tree in Fig. D.2 identifies three subevents that are the immediate causes of the top event. Each subevent represents a physically different process by which a reactor explosion can occur. There are three subevents in the fault tree in Fig. D.2, (1) Concentration of SO_2 too high in reactor, (2) Temperature of reactor too high and (3) Pressure in reactor too high. Each is an out-of-tolerance condition that can be detected by a sensing device, i.e., (1) a flow meter for the reactant stream, (2) a temperature gauge for the reactor and (3) a pressure gauge for the reactor. We use the concept of probabilistic importance to determine the most likely cause and, hence, the optimal sensor location.

---

*In reference to Chapter 5, we are considering self propagating fault events in which there is sufficient time for system diagnosis.

D.1 Process Description and Fault Tree Description

A process flow sheet is given in Fig. D.1 for SO_2-O_2 feed conversion to SO_3. The SO_2-O_2 feed stream is to enter the reactor which supplies excessive oxygen and inert nitrogen. These two streams are heated by superheated steam in heat exchangers. The pressure of the superheated steam is much greater than the pressure of the reactant streams. The SO_2 oxidation reaction in the reactor is regarded as highly exothermic and homogeneous.

The top event of the fault tree in Fig. D.2 identifies the major causes of reactor explosion

(1) Concentration of SO_2 too high in reactor

(2) Temperature in reactor too high

(3) Pressure in reactor too high

and are represented by subtrees in Fig. D.2.

The only safety devices for the boiler and the reactor are the pressure relief valves PR1 and PR2. In Fig. D.2, there are two separate failure mechanisms identified for the control valves, (1) the primary mechanical failure of the valve itself, and (2) the command faults of either the controller or sensor failing to close or open the control valve.

D.2 Basic Event Data and Cutsets

We assume that the system is at steady state. We adopt Option 4 of the IMPORTANCE computer code (see Appendix A). We assign a mean time to failure and mean fault duration time in terms of a reference unit μ for each basic event in the Fig. D.2 fault tree (see Table D-1). The input and output of the IMPORTANCE computer code for each subtree

is given in Fig. D.3. The input consists of the basic event data and min cut sets. The output consists of the steady state rate of breakdown (or occurrence) of each subevent and the ranking of each basic event.

D.4 Modular Decomposition Property at Steady State

The expression given by properties P1 and P2 in Section 3.2.3.1, are for the steady state case

$$\underline{P1} \quad I_i^{BP,SS} = \frac{[g(1^M, \underline{A}) - g(0^M, \underline{A})][h^M(1_i, \underline{A}) - h^M(0_i, \underline{A})]/(\mu_i + \tau_i)}{\sum_{M} \sum_{j \in M} [g(1^M, \underline{A}) - g(0^M, \underline{A})][h^M(1_j, \underline{A}) - h^M(0_j, \underline{A})]/(\mu_j + \tau_j)} .$$

$$\underline{P2} \quad I_M^{BP,SS} = \sum_{i \in M} I_i^{BP,SS} . \qquad (D.1)$$

D.5 Optimal Sensor Location

We now evaluate expression (D.1) to determine the subevent most likely to cause system failure. The probability of the top event is given by

$$g(\underline{A}) = 1 - [1 - h_1^M(\underline{A})][1 - h_2^M(\underline{A})][1 - h_3^M(\underline{A})]$$

where the numbered subscripts refer to the subtrees in the Fig. D.2 fault tree. The limiting unavailability of each subtree is given in Table D-2. (see Fig. D.3)

TABLE D-2

Subtree	Limiting Unavailability	$g(1^M, \underline{A}) - g(0^M, \underline{A})$
1, Concentration too High	$h_1^M(\underline{A}) = 1.01 \times 10^{-2}$	.932
2, Temperature too High	$h_2^M(\underline{A}) = 5.85 \times 10^{-2}$	.979
3, Pressure Too High	$h_3^M(\underline{A}) = 1.00 \times 10^{-2}$	.931

We can then compute the importance of each subtree by expression (D.1)

$$I_1^{BP,SS} = \frac{(.932)(1.011)}{(.932)(1.011) + (.979)(1.355) + (.931)(9.51 \times 10^{-4})}$$

$$= \frac{.942}{2.270} = .415$$

$$I_2^{BP,SS} = \frac{1.327}{2.270} = .584$$

$$I_3^{BP,SS} = \frac{8.9 \times 10^{-4}}{2.270} = 3.9 \times 10^{-4}$$

The expression $\sum_{i \in M} [h^M(1_i, \underline{A}) - h^M(0_i, \underline{A})]/(\mu_i + \tau_i)$ is the steady state rate of breakdown for module M, is given in Fig. D.3, and is directly substituted into expression (D.1) to obtain the above importance rankings. We see that 58.4% of the time a reactor explosion is caused by the temperature of the reactants being too high and 41.5% of the time an explosion is caused by concentration of SO_2 being too high. Explosion due to pressure of the reactant stream being too high makes a negligible contribution. On the basis of the quantitative results a designer should first consider putting a temperature gauge on the reactor. Almost equal consideration should be given to a flow meter for the reactant stream. Note as in Appendix B, the basic event data is given on a relative rather than absolute basis.

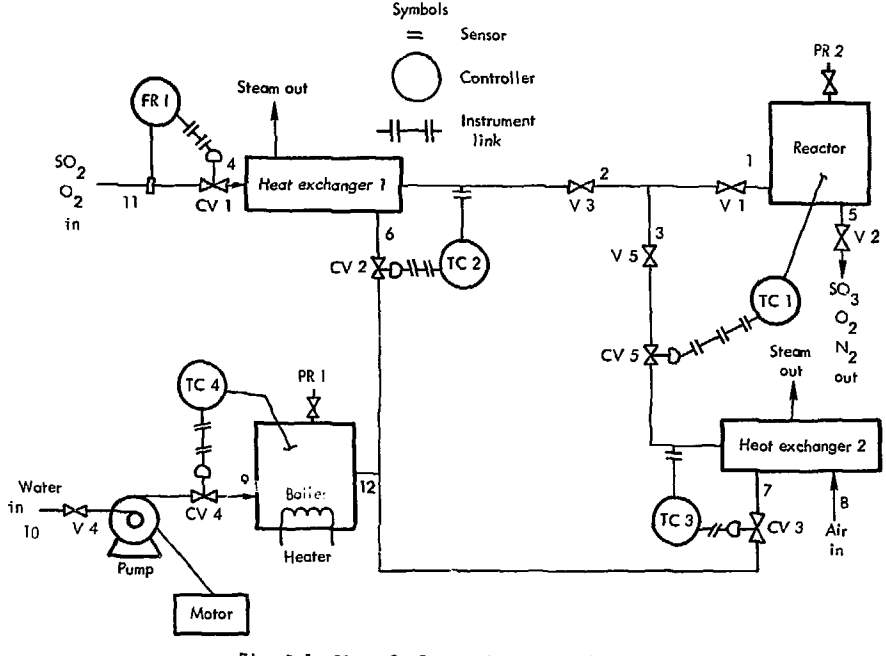


Fig. D.1 SO_2 - O_2 Conversion to SO_3 Process

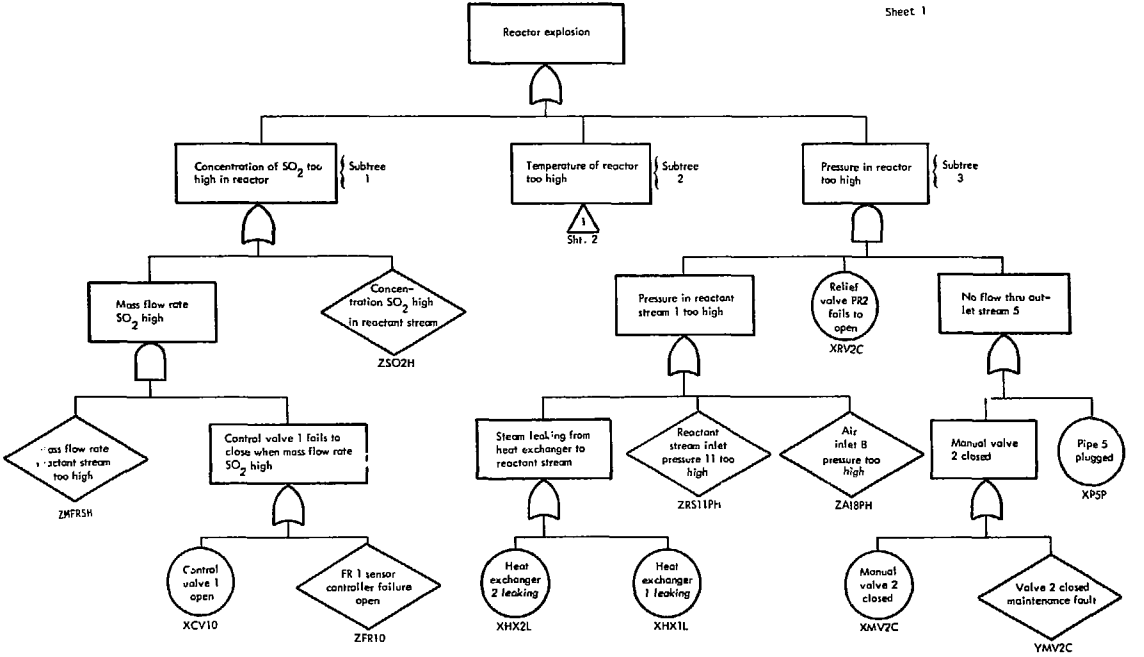


Fig. D.2 Fault Tree for Chemical Processing System

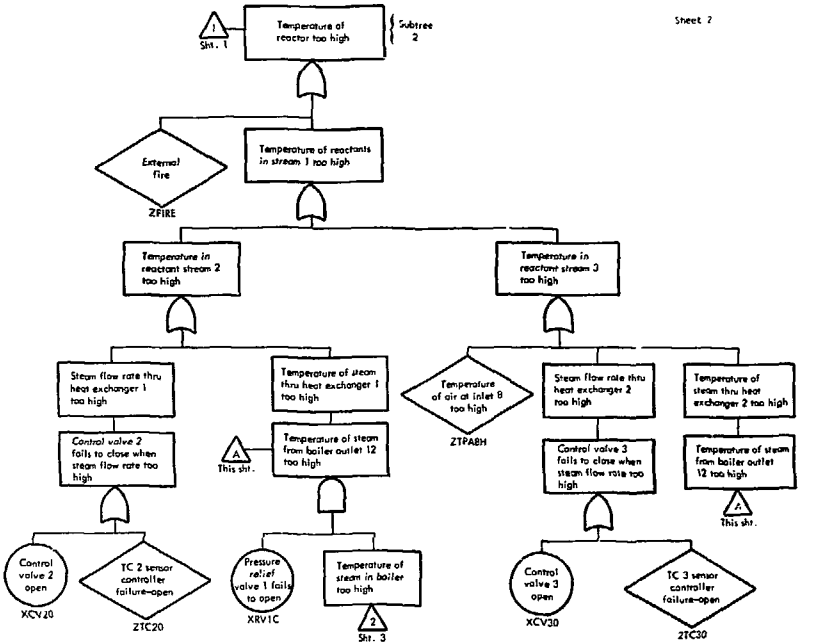

Sheet 2
1
Sht. 1
Temperature of reactor too high
Subtree 2
External fire
ZFIRE
Temperature of reactants in stream 1 too high
Temperature in reactant stream 2 too high
Temperature in reactant stream 3 too high
Steam flow rate thru heat exchanger 1 too high
Temperature of steam thru heat exchanger 1 too high
Temperature of air at inlet 8 too high
ZTPA8H
Steam flow rate thru heat exchanger 2 too high
Temperature of steam thru heat exchanger 2 too high
Control valve 2 fails to close when steam flow rate too high
A
This sht.
Temperature of steam from boiler outlet 12 too high
Control valve 3 fails to close when steam flow rate too high
Temperature of steam from boiler outlet 12 too high
A
This sht.
Control valve 2 open
XCV20
TC 2 sensor controller failure-open
ZTC20
Pressure relief valve 1 fails to open
XRV1C
Temperature of steam in boiler too high
2
Sht. 3
Control valve 3 open
XCV30
TC 3 sensor controller failure-open
ZTC30

Fig. D.2 Cont'd

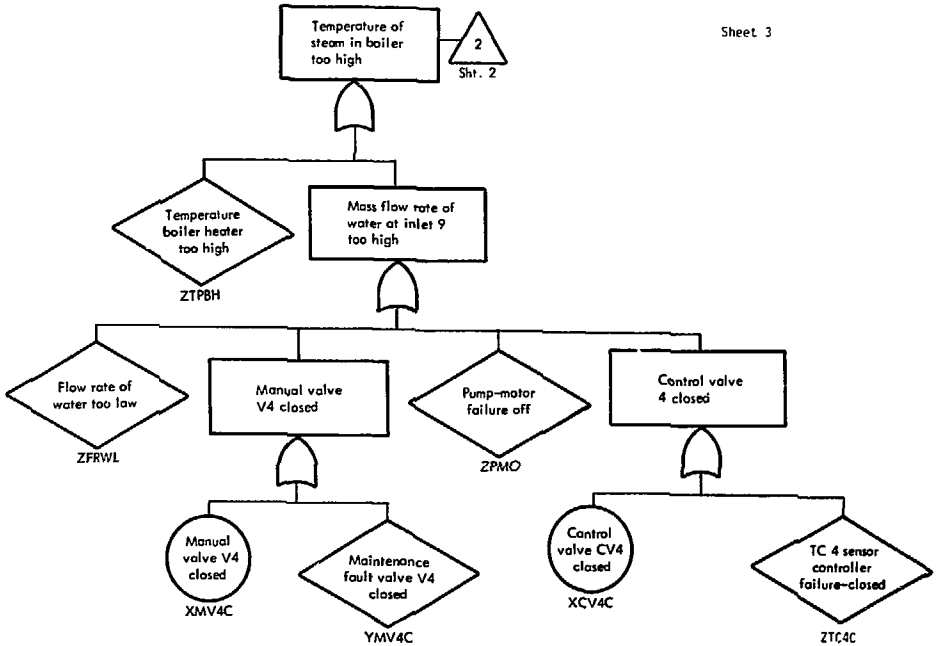

Temperature of steam in boiler too high
2
Sht. 2
Sheet 3
Temperature boiler heater too high
ZTPBH
Mass flow rate of water at inlet 9 too high
Flow rate of water too law
ZFRWL
Manual valve V4 closed
Pump-motor failure off
ZPMO
Control valve 4 closed
Manual valve V4 closed
XMV4C
Maintenance fault valve V4 closed
YMV4C
Control valve CV4 closed
XCV4C
TC 4 sensor controller failure-closed
ZTC4C

Fig. D.2 Cont'd

TABLE D-1

BASIC EVENT DATA

BASIC EVENT ALPHA-NUMERIC DESIGNATOR	BASIC EVENT DESCRIPTION	SUBTREE LOCATION	(Units of μ) MEAN TIME TO FAILURE	MEAN FAULT DURATION TIME
XCV1O	Control valve 1 open	1	10	.1
XCV2O	Control valve 2 open	2	10	.1
XCV3O	Control valve 3 open	2	10	.1
XCV4C	Control valve 3 closed	2	10	.1
XHX1L	Heat exchanger 1 leak	3	10	.1
XHX2L	Heat exchanger 2 leak	3	10	1
XMV2C	Manual valve 2 closed	3	10	.1
XMV4C	Manual valve 4 closed	2	10	.1
XP5P	Pipe 5 plugged	3	100	.1
XRV1C	Relief valve 1 fails to open	2	10	.1
XRV2C	Relief valve 2 fails to open	3	10	.1
YMV2C	Valve 2 closed maintenance fault	3	1	.01
YMV4C	Valve 4 closed maintenance fault	2	1	.01
ZAI8PH	Air inlet 8 pressure too high	3	1	01
ZFIRE	External fire	2	100	.1
ZFR1O	FR1 sensor controller fails to open	1	10	.1
ZFRWL	Flow rate of water too low	2	1	.01
ZMFRSH	Mass flow rate react. stream high	1	1	.01
ZPMO	Pump-motor failure off	2	10	.1
ZRS1OPH	Reactant stream inlet press. too high	3	1	.01
ZSO2H	Concentration SO_2 high in react. stream	1	1	.01
ZTC2O	TC 2 sensor controller fail-open	2	10	.1
ZTC3O	TC 3 sensor controller fail-open	2	10	.1
ZTC4C	TC 4 sensor controller fail. closed	2	10	.1
ZTPA8H	Temp. of air at inlet 8 too high	2	1	.01
ZTPBH	Temp. boiler heater too high	2	10	.1

```
      SUBTREE 1--CONCENTRATION OF SO2 TOO HIGH
          4          1
1.000E+02
          0          0          0          0          1          0
          0          0
          0          0
          0          0
          0     .00000
          4          3
          1  1.000E+00   .010E+00  ZMFRSH
          2  1.000E+01   .100E+00  XCV10
          3  1.000E+01   .100E+00  ZFR10
          4  1.000E+00   .010E+00  ZS02H
       4
       1     2
       1     3
```

INPUT

```
      SUBTREE 1--CONCENTRATION OF SO2 TOO HIGH
    STEADY STATE BREAKDOWN BASIC EVENT IMPORTANCE

    RATE OF SYSTEM BREAKDOWN AT STEADY STATE= 1.011E+00

    LIMITING SYSTEM UNAVAILIBILITY= 1.010E-02

RANK BASIC EVENT IMPORTANCE*

  1  ZS02H          9.790E-01*
  2  ZMFRSH         1.910E-02*
  3  ZFR10          9.590E-04*
  3  XCV10          9.590E-04*
```

OUTPUT

```
      SUBTREE 2--TEMPERATURE IN REACTOR TOO HIGH
          4          1
1.000E+02
          0          0          0          0          1          0
          0          0
          0          0
          0     .0
         14         12
          1  1.000E+02 1.000E+00  ZFIRE
          2  1.000E+01   .10E+00  XCV20
          3  1.000E+01   .10E+00  ZTC20
          4  1.000E+00   .01E+00  ZTPA8H
          5  1.000E+01   .10E+00  XCV30
          6  1.000E+01   .10E+00  ZTC30
          7  1.000E+01   .10E+00  XRV1C
          8  1.000E+01   .10E+00  ZTPBH
          9  1.000E+00   .01E+00  ZFRWL
         10  1.000E+01   .10E+00  XMV4C
         11  1.000E+00   .01E+00  YMV4C
         12  1.000E+01   .10E+00  ZPMP
         13  1.000E+01   .10E+00  XCV4C
         14  1.000E+01   .10E+00  ZTC4C
       1
       2
       3
       4
       5
       6
       7     8
       7     9
       7    10
       7    11
       7    12
       7    13
```

INPUT

Fig. D.3 Importance Listing for Chemical Processing System

```
          SUBTREE 2--TEMPERATURE IN REACTOR TOO HIGH
        STEADY STATE BREAKDOWN BASIC EVENT IMPORTANCE

        RATE OF SYSTEM BREAKDOWN AT STEADY STATE= 1.355E+00

        LIMITING SYSTEM UNAVAILIBILITY= 5.851E-02

RANK BASIC EVENT IMPORTANCE=

   1  ZTPA8H      5.940E-01=
   2  ZTC3D       6.948E-02=
   2  ZTC2D       6.948E-02=
   2  XCV2D       6.948E-02=
   2  XCV3D       6.948E-02=
   3  ZFIRE       6.948E-03=
   4  ZFRWL       6.812E-03=
   4  YMV4C       6.812E-03=
   5  XRV1C       3.909E-03=
   6  ZTPBH       6.812E-04=
   6  XMV4C       6.812E-04=
   6  ZPM5        6.812E-04=
   6  XCV4C       6.812E-04=
```

OUTPUT

```
          SUBTREE 3 --PRESSURE IN REACTOR TOO HIGH
                  4          1
1.000E+02
                  0          0          0          0          1          0
                  0          0
                  0          0
                  0       .0000D
                  8         12
                  1  1.000E+01  .100E+00  XRV1C
                  2  1.000E+01  .100E+00  XHX2L
                  3  1.000E+01  .100E+00  XHX1L
                  4  1.000E+00  .010E+00  ZRS11PH
                  5  1.000E+00  .010E+00  ZAI8PH
                  6  1.000E+01  .100E+00  XMV2C
                  7  1.000E+00  .010E+00  YMV2C
                  8  1.000E+02  .100E+00  XP5P
            1     2        6
            1     2        7
            1     2        8
            1     3        6
            1     3        7
            1     3        8
            1     4        6
            1     4        7
            1     4        8
            1     5        6
            1     5        7
            1     5        8
```

INPUT

```
          SUBTREE 3 --PRESSURE IN REACTOR TOO HIGH
        STEADY STATE BREAKDOWN BASIC EVENT IMPORTANCE

        RATE OF SYSTEM BREAKDOWN AT STEADY STATE= 9.514E-04

        LIMITING SYSTEM UNAVAILIBILITY= 1.001E-02

RANK BASIC EVENT IMPORTANCE=

   1  YMV2C       4.039E-01=
   2  ZAI8PH      2.122E-01=
   2  ZRS11PH     2.122E-01=
   3  XMV2C       4.039E-02=
   4  XRV1C       8.484E-02=
   5  XHX2L       2.122E-02=
   5  XHX1L       2.122E-02=
   6  XP5P        4.076E-03=
```

OUTPUT

Fig. D.3 Cont'd

APPENDIX E

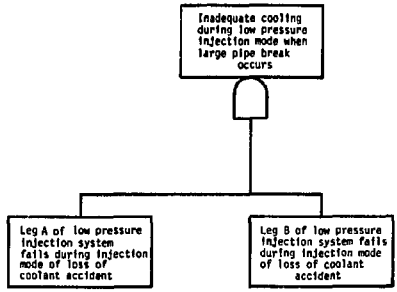
CHECKLIST GENERATION FOR LOW PRESSURE INJECTION SYSTEM

The model system that is chosen for checklist generation is a low pressure injection system (LPIS) at a pressurized water reactor nuclear power plant. The LPIS is a standy safety system, part of the emergency core cooling system. The piping schematic is shown in Fig. E.1. In the event of a loss of coolant accident, the low pressure injection system is designed to deliver 3000 gpm of water at 300 psi through each leg of the LPIS. The LPIS operation is considered successful when at least one leg of the LPIS discharges water continously at a rate of 3000 gpm into the cold legs of the reactor under accident conditions. This rate of injection is necessary to achieve adequate cooling of the core to prevent a fuel meltdown.

Part of the control circuit that actuates the LPIS is shown in Fig. E.2 which includes a brief description. For simplicity it is assumed that this system is tested by closing a switch (not shown) that energize relays that in turn close contacts P1, P2 and P3.

A checklist is generated for leg A in the event that the pressure gauge fails to indicate 300 psi when the test switch is closed. The checklist is a list of events that are most likely to have occurred when the leg A pressure gauge fails to indicate 300 psi as the test switch is closed.

The fault tree that simulates failure of the LPIS system is shown in Fig. E.3. It is part of a larger fault tree given below that describes failure of the entire LPIS.



Fault Tree for LPIS Failure

The basic events in the LPIS fault tree are coded according to a seven digit alphanumeric designator as was done in the Reactor Safety Study. The first digit indicates the event type, X represents a circle and Z represents a diamond. The second and third digits indicate the component type, e.g., PM stands for pump, MV for motor operated valve. The fifth and sixth digits identify for the specific events or components listed in the event description as given in Table E.1. The seventh digit represents the failure mode of the component, e.g., Q stands for short-circuit, A stands for "does not start", etc. Note that in the Fig. E.3 fault tree, human error is indicated as a cause of LPIS failure. For example, the basic event ZXVØ1Y located on the bottom of sheet 1 represents the event the "operator (or maintenance crew) inadvertently closes the manual valve".

The unavailabilities of all the basic events in the fault tree in Fig. E.3 are listed in Table E.1. As described in section 5.1.1.1, the unavailability of all active components required to change state is given by its cyclic failure rate. The emergency power buses are continuously operating systems; their unavailabilities are given by their limiting

asymptotic value, $\bar{A}$. The unavailability of all other components is given by the product $\lambda\tau$, where λ is an hourly failure rate and τ is the effective exposure time or fault duration time. The possibility of maintenance of these components is allowed as the entire LPIS is tested.

The Vesely-Fussell measure of importance of each event is given in Table E.2. It is identified that pump A failing to start has the greatest probability of causing failure of the LPIS to start. If pump A is working satisfactorily, then the first iteration in Table E.3 tells us that we should check the circuit breaker for pump A. During the second iteration we see that the basic events 23 and 30 are of equal importance and should be checked next. We continue this manner to generate a checklist of events. Fig. E.5 indicates the order in which the LPIS should be checked. This is the same as the initial listing as in Table E.2. The author generated other fault trees where the ordering of the basic events in the checklist was not the same as the ordering in the initial listing. In general, the iteration process is necessary for ordering basic events on the checklist.

The second iteration involves failure of a quasi-static component, i.e. a cable failure. At this point it is decided to check for a false alarm. In general a failure of an active component, i.e. a pressure gauge, is more likely to occur than a failure of a passive component.

During the seventh iteration, we start checking for components in the second order cut sets. A sublist for the motor operated valve #1 is generated in Fig. E.6. It is simply a listing of basic events contained in the same cut sets as MOV #1, ordered according to their probability of occurrence.

TABLE E-1

BASIC EVENT DATA FOR LPIS LEG A FAULT TREE

EVENT	NO.	SHEET	EVENT DESCRIPTION		UNAVAILABILITY†
XCB01K*	1	2	Cct. bkr. c acts fail to close		10^{-3}/d
XCV01C*	2	1	Check valve A jammed closed		10^{-4}/d
XCV02C*	3	1	Check valve B jammed closed		10^{-4}/d
XMV01D	4	1	Mov 1 fails to open	Hardware	1×10^{-3}/d
				Maintenance	$3.0\times10^{-3}\Delta t/T_M$
				Σ =	4.0×10^{-3}
XMV02D	5	1	Mov 2 fails to open	Hardware	1×10^{-3}/d
				Maintenance	$3.0\times10^{-3}\Delta t/T_M$
				Σ =	4.0×10^{-3}
XPD01K	6	5	Press. transd. contacts P1 fail to close		3×10^{-3}/d
XPD02K	7	5	Press. transd. contacts P2 fail to close		3×10^{-3}/d
XPD03K	8	5	Press. transd. contacts P3 fail to close		3×10^{-3}/d
XPM01A*	9	1	Pump Motor fails to start	Hardware	1×10^{-3}/d
				Maintenance	$2.5\times10^{-3}\Delta t/T_M$
				Σ =	3.5×10^{-3}
XRE01K*	10	2	#1 contacts fail to close		10^{-4}/d
XRE02K*	11	2	#4 contacts fail to close		10^{-4}/d
XRE03K	12	3	#2 contacts fail to close		10^{-4}/d
XRE04K	13	3	#5 contacts fail to close		10^{-4}/d
XRE05K**	14	3,4	#7 contacts fail to close		10^{-4}/d
XRE06K	15	4	#3 contacts fail to close		10^{-4}/d
XRE07K	16	4	#6 contacts fail to close		10^{-4}/d

†"d" represents "demand"; "$\Delta t/T_M$" represents "the fractional downtime due to maintenance"

TABLE E-1 Cont'd

EVENT	NO.	SHEET	EVENT DESCRIPTION	FAULT DURATION TIME (hrs), τ	FAILURE RATE λ	UNAVAIL-ABILITY†
XXV01D**	17	1	Manual valve fails to open		10^{-4}/d	10^{-4}
ZBS01N*	18	2	No power on bus 480 1H			$5\times10^{-4}(\bar{A})$
ZBS02N*	19	3,4,5	No power on bus DC1A			$5\times10^{-6}(\bar{A})$
ZBS03N*	20	4,5	No power on bus MCC1H1-1			$5\times10^{-4}(\bar{A})$
ZCB02O	21	3	Cct. bkr #1 open	720	10^{-6}/hr	7.2×10^{-4}
ZCB03O	22	4	Cct. bkr #2 open	720	10^{-6}/hr	7.2×10^{-4}
ZMV03C	23	1	N.O. MOV 3 inadvertantly closes	720	10^{-6}/hr	7.2×10^{-4}
ZPP01P*	24	1	Piping in leg A plugged	8760/2	10^{-8}/hr	4.4×10^{-5}
ZPP02P**	25	1	Piping from RWST plugged	8760/2	10^{-8}/hr	4.4×10^{-5}
ZPP01R*	26	1	Rupture in leg A of LPIS	720	10^{-7}/hr	7.2×10^{-5}
ZPP02R**	27	1	Rupture in pipe from RWST	720	10^{-7}/hr	7.2×10^{-5}
ZTR01O	28	3	Open cct. or short cct. transf. #1	720	10^{-6}/hr	7.2×10^{-4}
ZTR02O	29	4	Open cct. or short cct. transf. #2	720	10^{-6}/hr	7.2×10^{-4}
ZWR01O*	30	2	O.C. or S.C. in cable from LPI. PP. to bus 4801H	720	10^{-6}/hr	7.2×10^{-4}
ZWR02O	31	2	O.C. or S.C in wiring of close coil cct.	720	10^{-8}/hr	7.2×10^{-6}
ZWR03O*	32	2	O C. or S.C. in wiring of K1 coil cct.	720	10^{-8}/hr	$7\ 2\times10^{-6}$
ZWR04O*	33	2	O.C. or S.C. in wiring of K4 coil cct.	720	10^{-8}/hr	$7\ 2\times10^{-6}$
ZWR05O	34	3	O.C. or S.C. in cable from MOV-1 to bus MCC1H1-1	720	10^{-6}/hr	7.2×10^{-4}
ZWR06O	35	3	O.C. or S.C. in wiring of K2 coil cct.	720	10^{-8}/hr	7.2×10^{-6}
ZWR07O	36	3	O.C. or S.C. in wiring of K5 coil cct.	720	10^{-8}/hr	7.2×10^{-6}
ZWR08O**	37	4	O.C. or S.C. in wiring of K7 coil cct.	720	10^{-8}/hr	7.2×10^{-6}
ZWR09O	38	4	O.C. or S.C. in cable from MOV-2 to bus MCC1H1-1	720	10^{-6}/hr	7.2×10^{-4}
ZWR10O	39	4	O.C. or S.C. in wiring of K2 coil cct.	720	10^{-8}/hr	7.2×10^{-6}
ZWR11O	40	4	O.C. or S.C. in wiring of K6 coil cct.	720	10^{-8}/hr	7.2×10^{-6}
ZXV01Y**	41	1	Maintenance crew inadv. closes manual valve		10^{-4}/d	10^{-4}

†"$\bar{A}$" represents "limiting unavailability"

TABLE E-2 Importance Listing for Low Pressure Injection System

PROBABILITY THAT LPIS FAILS+ = 5.064E-04

PROBABILITY THAT LEG A FAILS+ = 7.956E-03

EVENT NO.	EVENT	UNAVAIL	IMPORTANCE	PROBABILITY LEG A FAILS+	PROBABILITY LPIS FAILS+
1	XCB01K*	1.0E-03	1.246E-01	1.000E+00	7.949E-03
2	XCV01C*	1.0E-04	1.247E-02	1.000E+00	7.949E-03
3	XCV02C*	1.0E-04	1.247E-02	1.000E+00	7.949E-03
4	XMV01D	4.0E-03	3.166E-03	1.422E-02	5.534E-04
5	XMV02D	4.0E-03	3.166E-03	1.422E-02	5.534E-04
6	XPD01K	3.0E-03	2.208E-03	1.387E-02	6.468E-03
7	XPD02K	3.0E-03	2.208E-03	1.387E-02	6.468E-03
8	XPD03K	3.0E-03	2.208E-03	1.387E-02	6.468E-03
9	XPM01A*	3.5E-03	4.380E-01	1.000E+00	7.949E-03
10	XRE01K*	1.0E-04	1.247E-02	1.000E+00	7.949E-03
11	XRE02K*	1.0E-04	1.247E-02	1.000E+00	7.949E-03
12	XRE03K	1.0E-04	7.885E-05	1.422E-02	5.534E-04
13	XRE04K	1.0E-04	7.885E-05	1.422E-02	5.534E-04
14	XRE05K**	1.0E-04	1.247E-02	1.000E+00	1.000E+00
15	XRE06K	1.0E-04	7.885E-05	1.422E-02	5.534E-04
16	XRE07K	1.0E-04	7.885E-05	1.422E-02	5.534E-04
17	XXV01D**	1.0E-04	1.247E-02	1.000E+00	1.000E+00
18	ZBS01N*	5.0E-04	6.238E-02	1.000E+00	7.949E-03
19	ZBS02N*	5.0E-06	6.235E-04	1.000E+00	7.949E-03
20	ZBS03N*	5.0E-04	6.238E-02	1.000E+00	7.949E-03
21	ZCB020	7.2E-04	5.680E-04	1.422E-02	5.534E-04
22	ZCB030	7.2E-04	5.680E-04	1.422E-02	5.534E-04
23	ZMV03C*	7.2E-04	8.984E-02	1.000E+00	7.949E-03
24	ZPP01P*	4.4E-05	5.487E-03	1.000E+00	7.949E-03
25	ZPP02R**	4.4E-05	5.487E-03	1.000E+00	1.000E+00
26	ZPP01R*	7.2E-05	8.978E-03	1.000E+00	7.949E-03
27	ZPP02R**	7.2E-05	8.978E-03	1.000E+00	1.000E+00
28	ZTR010	7.2E-04	5.680E-04	1.422E-02	5.534E-04
29	ZTR020	7.2E-04	5.680E-04	1.422E-02	5.534E-04
30	ZWR010*	7.2E-04	8.984E-02	1.000E+00	7.949E-03
31	ZWR020*	7.2E-06	8.978E-04	1.000E+00	7.949E-03
32	ZWR030*	7.2E-06	8.978E-04	1.000E+00	7.949E-03
33	ZWR040*	7.2E-06	8.978E-04	1.000E+00	7.949E-03
34	ZWR050	7.2E-04	5.680E-04	1.422E-02	5.534E-04
35	ZWR060	7.2E-06	5.676E-06	1.422E-02	5.534E-04
36	ZWR070	7.2E-06	5.676E-06	1.422E-02	5.534E-04
37	ZWR080**	7.2E-06	8.978E-04	1.000E+00	1.000E+00
38	ZWR090	7.2E-04	5.680E-04	1.422E-02	5.534E-04
39	ZWR100	7.2E-06	5.676E-06	1.422E-02	5.534E-04
40	ZWR110*	7.2E-06	5.676E-06	1.422E-02	5.534E-04
41	ZXV01Y**	1.0E-04	1.247E-02	1.000E+00	1.000E+00

LIST OF IMPORTANCE OF EVENTS IN ASCENDING ORDER

NO.	EVENT	UNAVAIL	IMPORTANCE
9	XPM01A*	3.5E-03	4.380E-01
1	XCB01K*	1.0E-03	1.246E-01
23	ZMV03C*	7.2E-04	8.984E-02
30	ZWR010*	7.2E-04	8.984E-02
18	ZBS01N*	5.0E-04	6.238E-02
20	ZBS03N*	5.0E-04	6.238E-02
2	XCV01C*	1.0E-04	1.247E-02
3	XCV02C*	1.0E-04	1.247E-02
10	XRE01K*	1.0E-04	1.247E-02
11	XRE02K*	1.0E-04	1.247E-02
14	XRE05K**	1.0E-04	1.247E-02
17	XXV01D**	1.0E-04	1.217E-02
41	ZXV01Y**	1.0E-04	1.247E-02
26	ZPP01R*	7.2E-05	8.978E-03
27	ZPP02R**	7.2E-05	8.978E-03
24	ZPP01P*	4.4E-05	5.487E-03
25	ZPP02R**	4.4E-05	5.487E-03
4	XMV01D	4.0E-03	3.166E-03
5	XMV02D	4.0E-03	3.166E-03
6	XPD01K	3.0E-03	2.238E-03
7	XPD02K	3.0E-03	2.238E-03
8	XPD03K	3.0E-03	2.238E-03
31	ZWR020*	7.2E-06	8.978E-04
32	ZWR030*	7.2E-06	8.978E-04
33	ZWR040*	7.2E-06	8.978E-04
37	ZWR080**	7.2E-06	8.978E-04
19	ZBS02N*	5.0E-06	6.235E-04
21	ZCB020	7.2E-04	5.680E-04
22	ZCB030	7.2E-04	5.680E-04
28	ZTR010	7.2E-04	5.680E-04
29	ZTR020	7.2E-04	5.680E-04
34	ZWR050	7.2E-04	5.680E-04
38	ZWR090	7.2E-04	5.680E-04
12	XRE03K	1.0E-04	7.885E-05
13	XRE04K	1.0E-04	7.885E-05
15	XRE06K	1.0E-04	7.885E-05
16	XRE07K	1.0E-04	7.885E-05
35	ZWR060	7.2E-06	5.676E-06
36	ZWR070	7.2E-06	5.676E-06
39	ZWR100	7.2E-06	5.676E-06
40	ZWR110*	7.2E-06	5.676E-06

+UPON DEMAND

**A BASIC EVENT WHOSE OCCURRENCE CAN CAUSE THE LPIS TO FAIL UPON DEMAND

*A BASIC EVENT WHOSE OCCURRENCE CAN CAUSE LEG A TO FAIL UPON DEMAND

TABLE E-3 Iteration Process for LPIS Checklist Generation

FIRST ITERATION

NO.	EVENT	UNAVAIL	IMPORTANCE
1	XCB01K*	1.0E-08	2.229E-01
23	ZMV03C*	7.2E-04	1.604E-01
30	ZWR016*	7.2E-04	1.604E-01
18	ZBS01N*	5.0E-04	1.114E-01
20	ZBS03N*	5.0E-04	1.114E-01
2	XCV01C*	1.0E-04	2.227E-02
3	XCV02C*	1.0E-04	2.227E-02
10	XRE01K*	1.0E-04	2.227E-02
11	XRE02K*	1.0E-04	2.227E-02
14	XRE05K**	1.0E-04	2.227E-02
17	XXV01D**	1.0E-04	2.227E-02
41	ZXV01Y**	1.0E-04	2.227E-02
26	ZPP01R*	7.2E-05	1.603E-02
27	ZPP02R**	7.2E-05	1.603E-02
24	ZPP01P*	4.4E-05	9.796E-03
25	ZPP02R**	4.4E-05	9.796E-03
4	XMV01D	4.0E-03	5.653E-03
5	XMV02D	4.0E-03	5.653E-03
6	XPD01K	3.0E-03	3.995E-03
7	XPD02K	3.0E-03	3.995E-03
8	XPD03K	3.0E-03	3.995E-03
31	ZWR020*	7.2E-06	1.603E-03
32	ZWR030*	7.2E-06	1.603E-03
33	ZWR040*	7.2E-06	1.603E-03
37	ZWR080**	7.2E-06	1.603E-03
19	ZBS02N*	5.0E-06	1.113E-03
21	ZCB02D	7.2E-04	1.014E-03
22	ZCB03D	7.2E-04	1.014E-03
28	ZTR01D	7.2E-04	1.014E-03
29	ZTR02D	7.2E-04	1.014E-03
34	ZWR05D	7.2E-04	1.014E-03
38	ZWR09D	7.2E-04	1.014E-03
12	XRE03K	1.0E-04	1.408E-04
13	XRE04K	1.0E-04	1.408E-04
15	XRE06K	1.0E-04	1.408E-04
16	XRE07K	1.0E-04	1.408E-04
35	ZWR06D	7.2E-06	1.013E-05
36	ZWR07D	7.2E-06	1.013E-05
39	ZWR10D	7.2E-06	1.013E-05
40	ZWR110*	7.2E-06	1.013E-05
9	XPM01A*	.0E+00	---------

SEVENTH ITERATION

NO.	EVENT	UNAVAIL	IMPORTANCE
4	XMV01D	4.0E-03	2.343E-01
5	XMV02D	4.0E-03	2.343E-01
6	XPD01K	3.0E-03	1.656E-01
7	XPD02K	3.0E-03	1.656E-01
8	XPD03K	3.0E-03	1.656E-01
31	ZWR020*	7.2E-06	6.643E-02
32	ZWR030*	7.2E-06	6.643E-02
33	ZWR040*	7.2E-06	6.643E-02
37	ZWR080**	7.2E-06	6.643E-02
19	ZBS02N*	5.0E-06	4.613E-02
21	ZCB02D	7.2E-04	4.203E-02
22	ZCB03D	7.2E-04	4.203E-02
28	ZTR01D	7.2E-04	4.203E-02
29	ZTR02D	7.2E-04	4.203E-02
34	ZWR05D	7.2E-04	4.203E-02
38	ZWR09D	7.2E-04	4.203E-02
12	XRE03K	1.0E-04	5.834E-03
13	XRE04K	1.0E-04	5.834E-03
15	XRE06K	1.0E-04	5.834E-03
16	XRE07K	1.0E-04	5.834E-03
35	ZWR06D	7.2E-06	4.200E-04
36	ZWR07D	7.2E-06	4.200E-04
39	ZWR10D	7.2E-06	4.200E-04
40	ZWR110*	7.2E-06	4.200E-04
1	XCB01K*	.0E+00	---------
2	XCV01C*	.0E+00	---------
3	XCV02C*	.0E+00	---------
9	XPM01A*	.0E+00	---------
10	XRE01K*	.0E+00	---------
11	XRE02K*	.0E+00	---------
14	XRE05K**	.0E+00	---------
17	XXV01D**	.0E+00	---------
18	ZBS01N*	.0E+00	---------
20	ZBS03N*	.0E+00	---------
23	ZMV03C*	.0E+00	---------
24	ZPP01P*	.0E+00	---------
25	ZPP02R**	.0E+00	---------
26	ZPP01R*	.0E+00	---------
27	ZPP02R**	.0E+00	---------
30	ZWR016*	.0E+00	---------
41	ZXV01Y**	.0E+00	---------

Components Checked and Verified O.K.

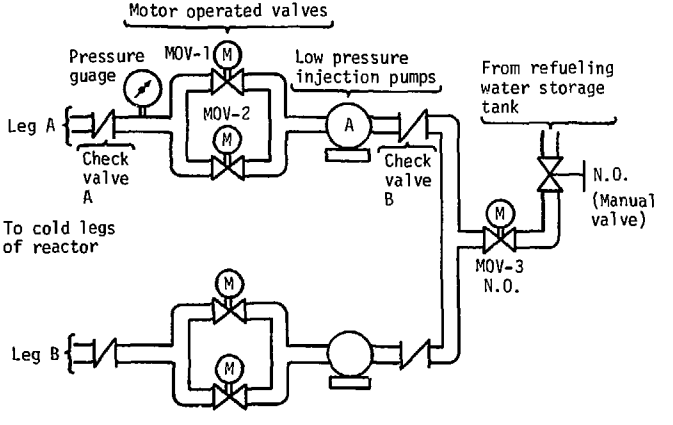


Fig. E.1

LOW PRESSURE INJECTION SYSTEM

The engineered safeguard systems including the LPIS are actuated by a safety injection signal, SIS. The loss of coolant accident create conditions, such as low pressurizer water level and low pressurizer pressure that are detectable by transducers. For simplicity, the actuation of the LPIS is described when the SIS is generated from the 2-out-of-3 circuit for high containment pressure.

In the event of high containment pressure, pressure transducers 1, 2 and 3 (not shown) close contacts P1, P2 and P3. DC current then energizes relay coil K7 and the #7 contacts close. In turn the interposing relays K4, K5 and K6 are energized. Then the #4, #5 and #6 contacts close and energize respectively relay coils K1, K2 and K3 that in turn close the #1, #2 and #3 contacts. The close coil to the circuit breaker of the LPI pump A closes its contacts that in turn provide 480 V 3 phase power to pump A. Similarly the #2 and #3 contacts close and provide 120 V power to the motors that open valves MOV-1 and MOV-2 respectively.

Fig. E.2
DESCRIPTION OF THE LPIS
CONTROL SYSTEM

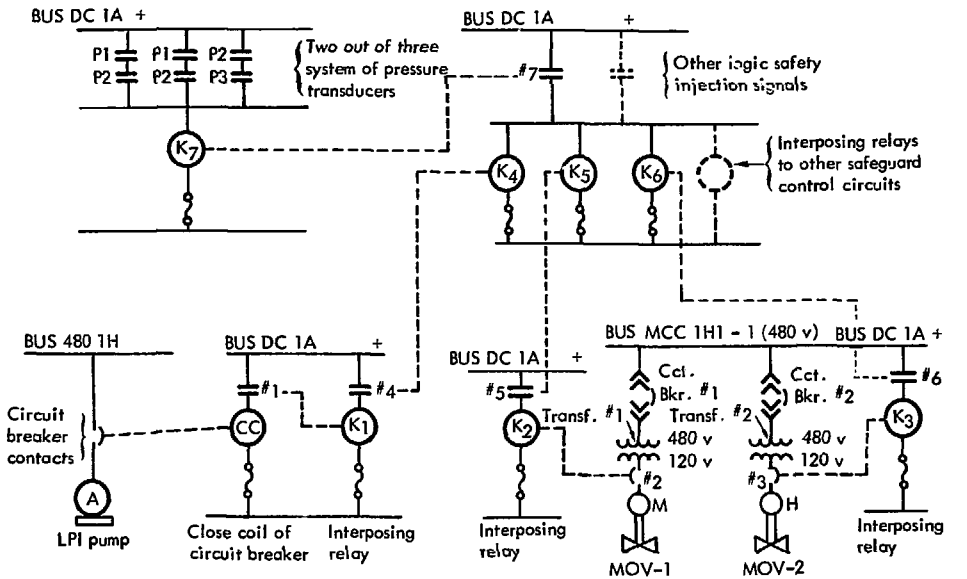


Fig. E.2 LPIS CONTROL CIRCUIT

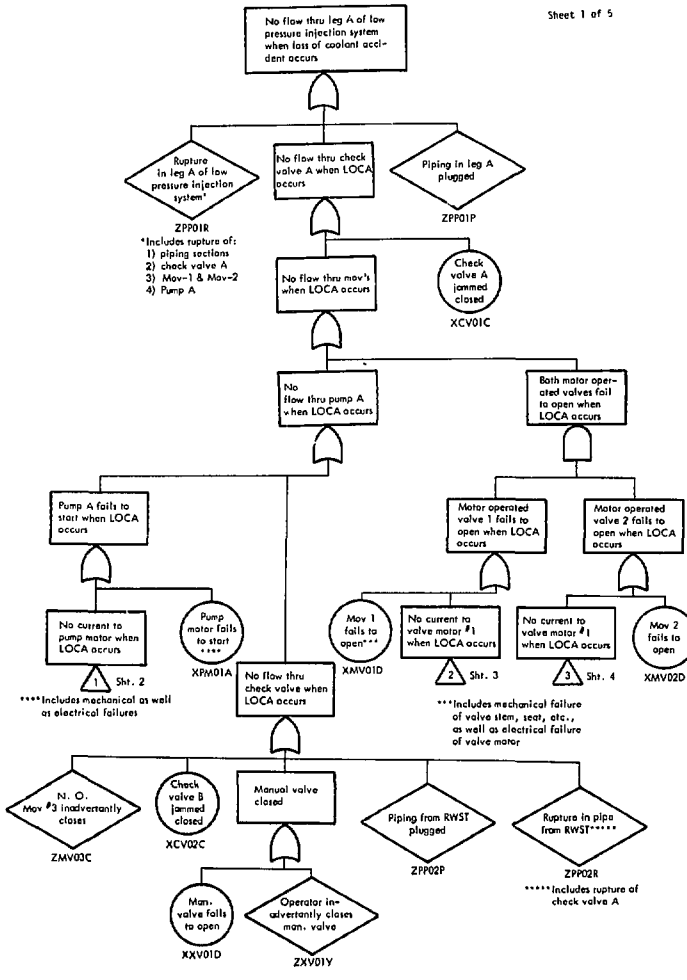


Fig. E.3 FAULT TREE FOR LOW PRESSURE INJECTION SYSTEM

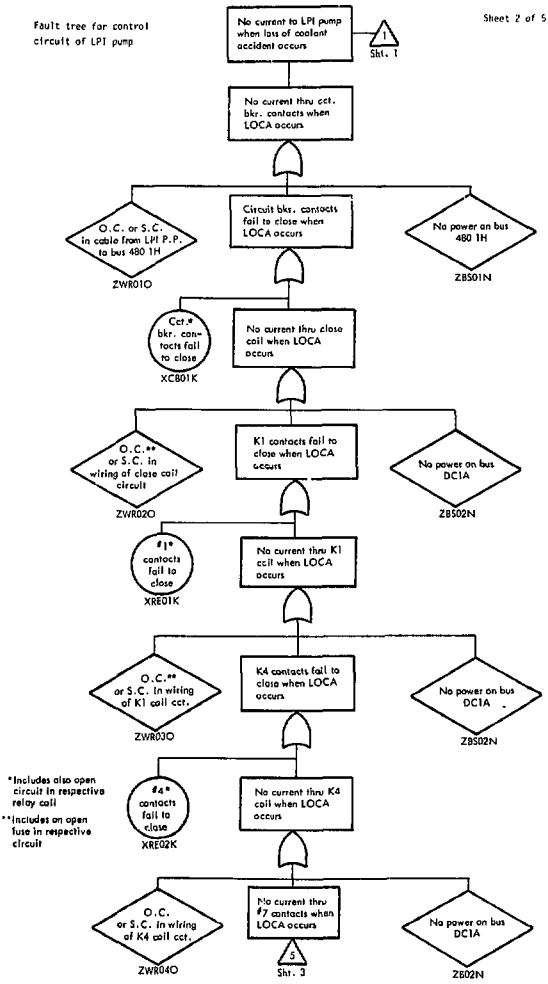

Fault tree for control circuit of LPI pump
Sheet 2 of 5
No current to LPI pump when loss of coolant accident occurs
1
Sht. 1
No current thru cct. bkr. contacts when LOCA occurs
O.C. or S.C. in cable from LPI P.P. to bus 480 1H
ZWR01O
Circuit bkr. contacts fail to close when LOCA occurs
No power on bus 480 1H
ZBS01N
Cct.# bkr. contacts fail to close
XCB01K
No current thru close coil when LOCA occurs
O.C.** or S.C. in wiring of close coil circuit
ZWR02O
K1 contacts fail to close when LOCA occurs
No power on bus DC1A
ZBS02N
#1* contacts fail to close
XRE01K
No current thru K1 coil when LOCA occurs
O.C.** or S.C. in wiring of K1 coil cct.
ZWR03O
K4 contacts fail to close when LOCA occurs
No power on bus DC1A
ZBS02N
*Includes also open circuit in respective relay coil
**Includes an open fuse in respective circuit
#4* contacts fail to close
XRE02K
No current thru K4 coil when LOCA occurs
O.C. or S.C. in wiring of K4 coil cct.
ZWR04O
No current thru #7 contacts when LOCA occurs
3
Sht. 3
No power on bus DC1A
ZB02N

Fig. E.3 Cont'd

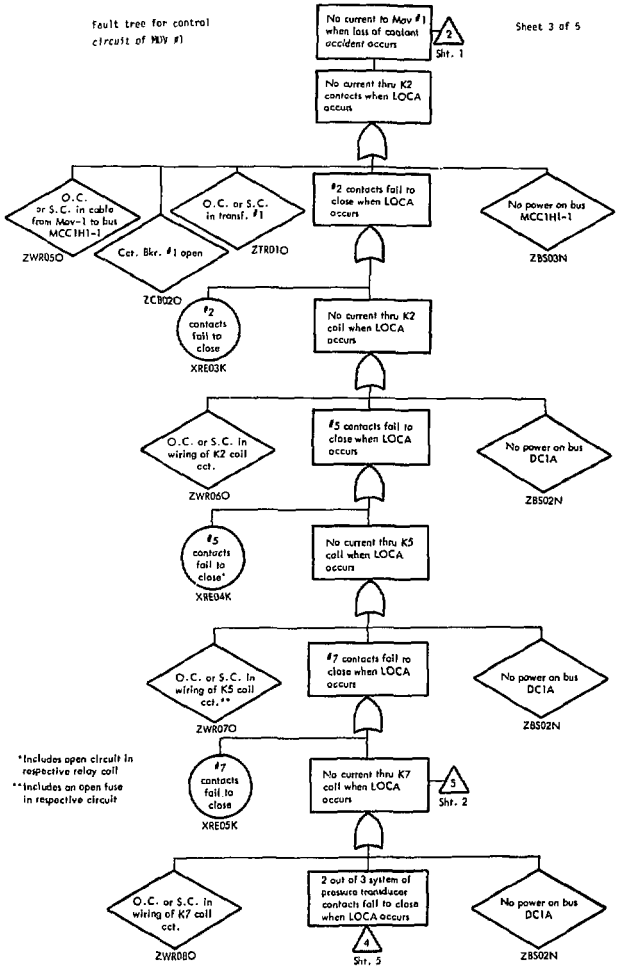

Fault tree for control circuit of MOV #1
Sheet 3 of 5
No current to Mov #1 when loss of coolant accident occurs
2
Sht. 1
No current thru K2 contacts when LOCA occurs
#2 contacts fail to close when LOCA occurs
O.C. or S.C. in cable from Mov-1 to bus MCC1H1-1
ZWR05O
O.C. or S.C. in transf. #1
ZTR01O
No power on bus MCC1H1-1
ZBS03N
Cct. Bkr. #1 open
ZCB02O
#2 contacts fail to close
XRE03K
No current thru K2 coil when LOCA occurs
#5 contacts fail to close when LOCA occurs
O.C. or S.C. in wiring of K2 coil cct.
ZWR06O
No power on bus DC1A
ZBS02N
#5 contacts fail to close*
XRE04K
No current thru K5 coil when LOCA occurs
#7 contacts fail to close when LOCA occurs
O.C. or S.C. in wiring of K5 coil cct.**
ZWR07O
No power on bus DC1A
ZBS02N
*Includes open circuit in respective relay coil
**Includes an open fuse in respective circuit
#7 contacts fail to close
XRE05K
No current thru K7 coil when LOCA occurs
3
Sht. 2
2 out of 3 system of pressure transducer contacts fail to close when LOCA occurs
4
Sht. 5
O.C. or S.C. in wiring of K7 coil cct.
ZWR08O
No power on bus DC1A
ZBS02N

Fig. E.3 Cont'd

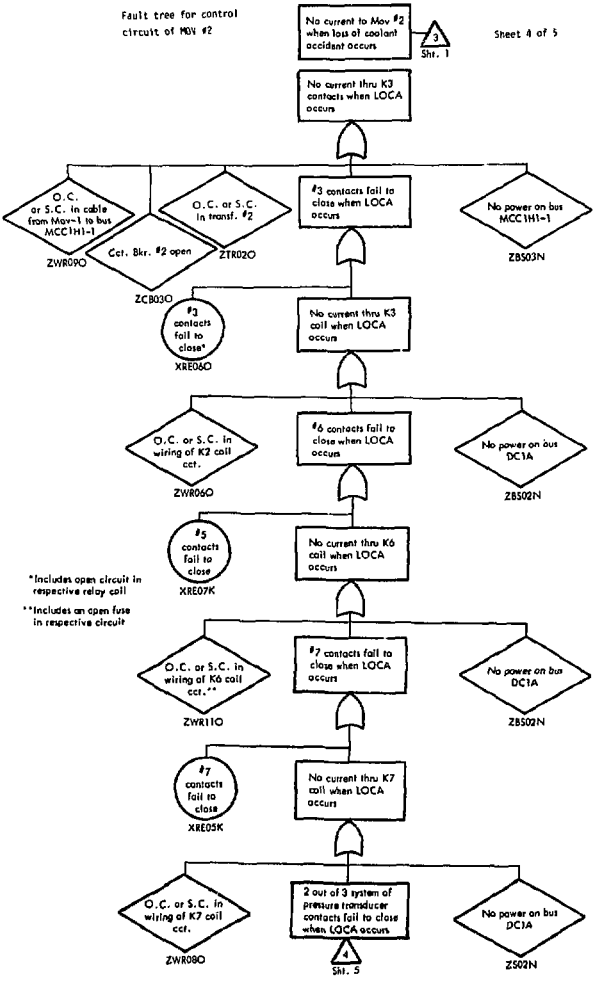
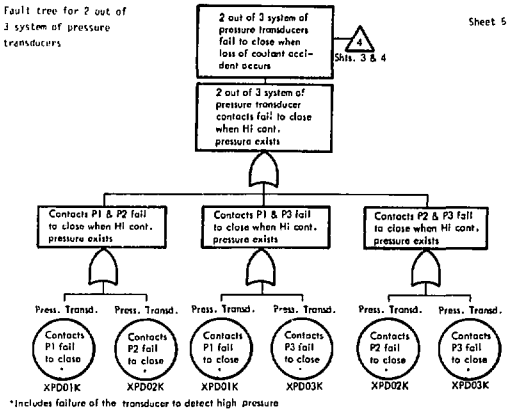
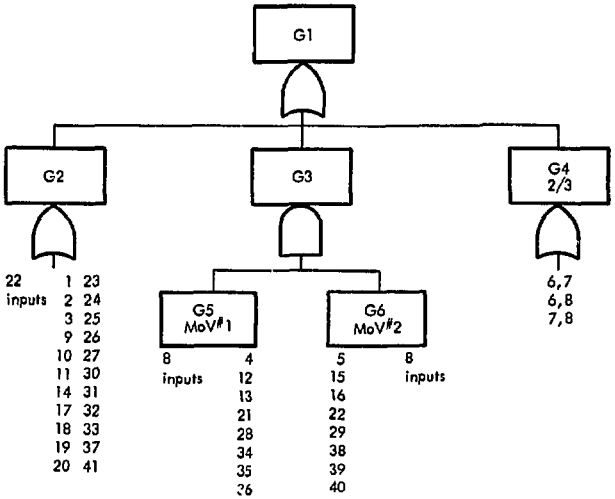

Fault tree for control circuit of MOV #2
No current to Mov #2 when loss of coolant accident occurs
3 Shr. 1
Sheet 4 of 5
No current thru K3 contacts when LOCA occurs
O.C. or S.C. in cable from Mov-1 to bus MCC1H1-1
ZWR09O
Ckt. Bkr. #2 open
ZCB03O
O.C. or S.C. in transf. #2
ZTR02O
#3 contacts fail to close when LOCA occurs
No power on bus MCC1H1-1
ZBS03N
#3 contacts fail to close*
XRE06O
No current thru K3 coil when LOCA occurs
O.C. or S.C. in wiring of K2 coil ckt.
ZWR06O
#6 contacts fail to close when LOCA occurs
No power on bus DC1A
ZBS02N
#5 contacts fail to close
XRE07K
No current thru K6 coil when LOCA occurs
*Includes open circuit in respective relay coil
**Includes an open fuse in respective circuit
O.C. or S.C. in wiring of K6 coil ckt.**
ZWR11O
#7 contacts fail to close when LOCA occurs
No power on bus DC1A
ZBS02N
#7 contacts fail to close
XRE05K
No current thru K7 coil when LOCA occurs
O.C. or S.C. in wiring of K7 coil ckt.
ZWR08O
2 out of 3 system of pressure transducer contacts fail to close when LOCA occurs
4 Shr. 5
No power on bus DC1A
ZS02N

Fig. E.3 Cont'd



*Includes failure of the transducer to detect high pressure

Fig. E.3 Cont'd



22 1st order cut sets
8·8 + 3 = 67 2nd order cut sets
89 cut sets TOTAL

BOOLEAN EXPRESSION FOR LPIS FAULT TREE

$$G2 = 1-(1-y_1)(1-y_2)\cdot\ \ldots\ \cdot(1-y_{41})$$

22 inputs

$$G5 = 1-(1-y_4)(1-y_{12})\cdot\ \ldots\ \cdot(1-y_{36})$$

8 inputs

$$G6 = 1-(1-y_5)(1-y_{15})\cdot\ \ldots\ \cdot(1-y_{40})$$

8 inputs

$$G4 = 1-(1-y_6\cdot y_7)(1-y_6\cdot y_8)(1-y_7\cdot y_8)$$

$$= y_6\cdot y_7\cdot y_8 + (1-y_6)y_7\cdot y_8 + y_6(1-y_7)\cdot y_8 + y_6\cdot y_7(1-y_8)$$

$$\psi(\underline{y}) = G1 = 1-(1-G2)(1-G5\cdot G6)(1-G4)$$

BOOLEAN EXPRESSION FOR TOP EVENT

Fig. E.4 BOOLEAN EQUIVALENT OF THE LPIS FAULT TREE

ORDER	EVENT NO.	EVENT
1	9	XPM01A*
2	1	XCB01K*
3	23	ZMV03C
CHECK FOR FALSE ALARM		
3	30	ZWR01O*
4	18	ZBS01N*
	20	ZBS03N*
5	2	XCV01C*
	3	XCV02C*
	10	XRE01K*
	11	XRE02K*
	14	XRE05K**
	17	XXV01D**
	41	ZXV01Y**
6	26	ZPP01R*
	27	ZPP02R**
7	24	ZPP01P*
	25	ZPP02R**
8	4	XMV01D
	5	XMV02D
9	6	XPD01K
	7	XPD02K
	8	XPD03K
10	31	ZWR02O*
	32	ZWR03O*
	33	ZWR04O*
	37	ZWR08O*
11	19	ZBS02N*
12	21	ZCB02O
	22	ZCB03O
	28	ZTR01O
	29	ZTR02O
	34	ZWR05O
	38	ZWR09O
13	12	XRE03K
	13	XRE04K
	15	XRE06K
	16	XRE07K
14	35	ZWR06O
	36	ZWR07O
	39	ZWR10O
	40	ZWR11O*

Fig. E.5 CHECKLIST FOR LEG A OF LPIS

Order in which the basic events on the LPIS fault tree should be checked

ORDER	EVENT NO.	EVENT
1	5	XMV02D
2	22	ZCB03O
	29	ZTR02O
	38	ZWR09O
3	15	XRE06K
	16	XRE07K
4	39	ZWR10O
	40	ZWR11O

Fig. E.6 SUBLIST FOR MOTOR OPERATED VALVE #1