

NUCLEAR ENGINEERING
READING ROOM - M.I.T.

MITNE-251

APPLICATION OF
TIME DEPENDENT UNAVAILABILITY ANALYSIS
TO STANDBY SAFETY SYSTEMS

Andrew Arthur Dykes
Massachusetts Institute of Technology

Norman C. Rasmussen
Massachusetts Institute of Technology

William E. Vesely, Jr.
Battelle Memorial Institute

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Department of Nuclear Engineering

Cambridge, Massachusetts 02139

June 1982

Prepared for Brookhaven National Laboratory
under Contract No. BNL-546681

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Nuclear Regulatory Commission, nor any of their employees, makes any warranty, express or implied, nor assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, nor represents that its use would not infringe privately owned rights.

APPLICATION OF
TIME DEPENDENT UNAVAILABILITY ANALYSIS
TO STANDBY SAFETY SYSTEMS

by

ANDREW ARTHUR DYKES

Submitted to the Department of Nuclear Engineering
on May 19, 1982 in Partial Fulfillment of the
Requirements for the Degree of Doctorate in Philosophy

ABSTRACT

The FRANTIC II computer code has been modified and used to demonstrate that time dependent unavailability analysis is a practical tool for assessing the periodic testing programs of operational standby safety systems.

FRANTIC II was assessed from an engineering point of view and modified as necessary to make it more useful for application to operational systems. An offset time was added to the component failure parameters to provide more flexible modeling of time dependent standby failures and the effects of test caused wear-out. A routine to calculate the optimum test interval of a constant failure rate component subject imperfect testing was also developed. The code was then coupled to a cutset generator and evaluator for application to multiple component systems. The resulting code is named FRANTIC II-MIT.

FRANTIC II-MIT has been applied to the High Pressure Coolant Injection System of a Boiling Water Reactor and a quantitatively based periodic testing program keyed to a fault tree evaluation of the system's safety functions has been formulated. The analysis indicated that system unavailability can be reduced while also reducing testing requirements from approximately 170 to 123 tests per year.

Thesis Supervisor: Norman C. Rasmussen
Title: Professor of Nuclear Engineering

Thesis Supervisor: William E. Vesely, Jr.
Title: Research Affiliate

TABLE OF CONTENTS

| | | |
|-----------|--|----|
| Chapter 1 | INTRODUCTION | 1 |
| 1.1 | Background and Motivation | 3 |
| 1.2 | Purpose and Scope | 7 |
| 1.3 | Structure of Thesis | 10 |
| Chapter 2 | UNAVAILABILITY AND PERIODIC TESTING | 13 |
| 2.1 | Standby System Unreliability Definitions | 13 |
| | Unavailability, $Q(t_1)$ | 14 |
| | Cumulative Failure Probability, $P(t_1, t_2)$ | 15 |
| | Unreliability, $F(t_1, t_2)$ | 17 |
| 2.2 | Basic Unavailability Concepts | 18 |
| | 2.2.1 "Other" Components | 18 |
| | 2.2.2 Periodically Tested Components | 22 |
| 2.3 | Current Status of Analysis of Periodic Testing | 25 |
| | 2.3.1 Regulations and Standards | 26 |
| | 2.3.2 Published Research | 29 |
| | Simple Systems With Test Downtime | 29 |
| | Redundant Standby Components | 36 |
| | Effects of Human Error on Simple Systems | 40 |
| | Components With Many Failure Mechanisms | 47 |
| | 2.3.3 Application to Fault Tree Analysis | 55 |
| | 2.3.4 The FRANTIC II Computer Code | 57 |

TABLE OF CONTENTS (Continued)

Chapter 3 ENGINEERING INTERPRETATION OF FRANTIC II-MIT . . 60

3.1 Overall Structure 62

 3.1.1 Unavailability Equations 62

 3.1.2 Calculation Procedure 65

3.2 Standby Failure Rate 71

 3.2.1 Scale Factor for Detectable Standby Failures . . . 73

 3.2.2 Scale Factor for Undetectable Standby Failures . . 74

 3.2.3 Failure Rate Shape Factor 75

 3.2.4 Component Renewal Type 76

 Use of Offset Time With Renewal Types 78

 3.2.5 Test Caused Changes to Scale Factor 78

 3.2.6 Unavailability Due to Standby Failures 79

 Monitored Components 79

 Periodically Tested Components 80

3.3 Demand Failure Rate 82

 3.3.1 Engineering Interpretation 82

 3.3.2 Special Uses 83

 Monitored Failures in Periodically Tested Components . . 83

 Common Cause Failures 84

3.4 Times Associated with Periodic Testing 84

 3.4.1 Periodic Inspection Interval 85

 3.4.2 First Periodic Inspection Interval 85

 3.4.3 Scheduled Test and Maintenance Period 85

 3.4.4 Unscheduled Repair Time 86

TABLE OF CONTENTS (Continued)

| | | |
|---|--|-----|
| 3.5 | Effects of Imperfect Testing | 87 |
| 3.5.1 | Probability of Test Caused Failure | 87 |
| 3.5.2 | Unavailability to Override Test and Maintenance | 88 |
| 3.5.3 | Test Caused Failure Rate Change Factors . . . | 89 |
| 3.5.4 | Test Error Carryover Factor | 92 |
| 3.6 | Generalized Weibull Hazard Rate | 93 |
| 3.6.1 | Susceptibility of Component Failure Mechanisms | 95 |
| 3.6.2 | Effects of Offset Time on Hazard Rate | 98 |
| 3.6.3 | Advantages Over FRANTIC II Hazard Rate | 102 |
| 3.6.4 | Estimating Input Parameters | 109 |
| 3.6.5 | Alternate Models of Time Dependency | 114 |
| Chapter 4 APPLICATION TO SINGLE COMPONENT SYSTEMS . . . | | 118 |
| 4.1 | Demand Verses Standby Failures | 118 |
| 4.1.1 | Ratio of Observed Demand and Standby Failures . | 119 |
| | Beta = 1 Standby Failure Rate | 121 |
| | Beta = 2 Standby Failure Rate, NN Renewal | 123 |
| 4.1.2 | Effect of Increasing Demand Failure Rate . . . | 126 |
| 4.1.3 | Effect of Undetectable Failures | 126 |
| 4.1.4 | FRANTIC Code Assumption Regarding Demand Failures | 129 |
| 4.2 | Subroutine OPTTEST | 130 |
| 4.2.1 | Background | 130 |
| 4.2.2 | Theory | 130 |

TABLE OF CONTENTS (Continued)

4.2.3 Implementation in FRANTIC II-MIT 138

4.2.4 Comparison With FRANTIC II-MIT RUN Option 139

4.3 Behavior of Constant Failure Rate Components 143

4.3.1 Approximate Relative Importance of Failure Modes . 143

4.3.2 Effective Test Downtime 145

4.3.3 Unavailability Contours 147

4.4 Behavior of Time Dependent Hazard Rate Components . . 149

4.4.1 Importance of Hazard Rate 149

4.4.2 Uses of Renewal Options 151

 New-New Renewal 151

 Old-New Renewal 152

 Old-Old Renewal 153

4.5 Test Caused Wear-Out and Burn-In 157

4.5.1 Effect on Standby Failure Rate 157

4.5.2 Effect on Demand Failure Rate 161

4.6 Summary 161

Chapter 5 APPLICATION TO MULTIPLE COMPONENT SYSTEMS . . . 163

5.1 CUTSETS 164

5.2 Some Applications to Simple Systems 170

5.2.1 Series Systems 170

5.2.2 Demand Failures and Redundant Components 172

5.2.3 Effect of Unequal Test Override 174

TABLE OF CONTENTS (Continued)

5.3 Comparison With Vaurio's 1-out-of-3 System
 Calculation 174
 System Description and Cut Sets 176

5.4 An Approach for Analyzing Systems 183

Chapter 6 APPLICATION TO A HIGH PRESSURE COOLANT
 INJECTION SYSTEM 186

6.1 Introduction 186
 Organization of Analysis 187

6.2 Description of System Functions , 189
 6.2.1 Safety Functions 190
 6.2.2 Injection Function 192
 Actions Required for Injection 192
 6.2.3 Autoisolation and Termination 195
 Actions Required for Autoisolation 198

6.3 Fault Tree Analysis 198
 6.3.1 Assumptions 199
 6.3.2 Qualitative Analysis 201
 6.3.3 Quantitative Analysis 205

6.4 Turbine/Pump Train Operability Tests 205
 6.4.1 Description of Online Tests 205
 Recommendations for Consolidation 207
 6.4.2 Description of Operating Cycle Tests 207

TABLE OF CONTENTS (Continued)

6.4.3 Quantitative Analysis of Operational Tests

- Super Component Failure Rate 209
- Steam Supply Valve Failure Rate 212
- Unavailability During Testing 215
- Results 217
- Recommendations 227

6.5 Automatic Initiation Function Tests 227

- 6.5.1 Initiation Sensor Tests 231
 - Quantitative Analysis 232
- 6.5.2 Description of Initiation Logic Tests 237
 - Current Test Policy 237
 - Comments on Current Test Procedures 238
- 6.5.3 Quantitative Evaluation of Initiation Logic Tests . 239
 - Unavailability Using Current Logic Design 241
 - Initiation Logic Relay Modifications 244
 - Summary and Recommendations 248

6.6 Autoisolation Function Tests 250

- 6.6.1 Quantitative Analysis 255
 - Common Cause Effects 255
 - Important Cut Sets 256
 - Interaction With Injection Function 260
 - Recommendations 263

6.7 Summary of HPCI Recommendations 266

TABLE OF CONTENTS (Continued)

Chapter 7 SUMMARY, CONCLUSIONS AND RECOMMENDATIONS . . 273
7.1 Summary 273
7.2 Conclusions 275
7.3 Recommendation for Further Research 277

APPENDICES

A HPCI Injection Function Fault Tree 285
B HPCI Injection Function Cut Sets 305
C HPCI Injection Function Components 312
D HPCI Injection Function Fault Tree Modifica- . 316
tions Resulting From Initiation Logic Changes
E HPCI Initiation Function Cut Sets (Before . . . 319
and After)
F HPCI Autoisolation Function Fault Tree 324
G HPCI Autoisolation Function Cut Sets 332
H HPCI Autoisolation Function Components 335
I FRANTIC-II Input 337
J The CUTSETS Package 355
K Running FRANTIC II-MIT on CMS 360

List of Figures

| <u>Figure</u> | <u>Description</u> | |
|---------------|---|----|
| 1.1 | Simplified Example of a Periodically Tested System. | 4 |
| 2.1 | Time Dependent Unavailability of Components Other Than Those That Must Be Periodically Tested to Reveal Standby Failures. | 19 |
| 2.2 | Time Dependent Unavailability of Components Which Must Be Tested to Reveal Standby Failures. | 23 |
| 2.3 | Example of the Effect of Test Interval on Component Unavailability When the Component is Completely Unavailable for its Safety Function While Being Tested. | 30 |
| 2.4 | Unavailability of a Component Subject to Standby Failures and Test Downtime. [Ja68] | 30 |
| 2.5 | Average Component Unavailability Verses Time Between Tests, Parametric With Component Failure Rate, λ (failures/hr), Outage Duration (τ) = 1 hour. (Figure 4 of NUREG/CR-2158) | 35 |
| 2.6 | Nomograph for Calculating a Test Interval and Duration to Meet an Unavailability Goal. Example for 2-out-of-3:Good Logic. [Hi71] | 39 |
| 2.7 | Optimal (Maximum Availability) Test Interval and Associated Availability as a Function of p_A . [McW80] | 42 |
| 2.8 | Optimal (Maximum Availability) Test Interval and Associated Availability as a Function of p_B . [McW80] | 42 |
| 2.9 | Test Interval of Diesel Generator Failures. [Ma82] | 49 |
| 3.1 | Computational Flow of the FRANTIC Computer Programs. | 66 |
| 3.2 | Example of FRANTIC's Use of Time Points to Calculate the Instantaneous Unavailability of a Two Component Parallel System. | 67 |

List of Figures

| <u>Figure</u> | <u>Description</u> | |
|---------------|---|-----|
| 3.3 | FRANTIC II-MIT Output of System Unavailibility Data Resulting From a Calculation Using the Run Option. | 70 |
| 3.4 | Use of Offset Time and Renewal Options to Obtain Time Dependent Failure Rates With a Generalized Weibull Hazard Rate. | 100 |
| 3.5 | Five Failure Rates Modeled With a Two Parameter Weibull Hazard Rate Which Rise From 0 at Time Zero to 1.0xE-5/hr at 20 Years. | 104 |
| 3.6 | Five Failure Rates Modeled With a Generalized Weibull Hazard Rate Which Rise From 1.0E-6/hr at Time Zero to 1.0E-5/hr at 20 Years. | 108 |
| 3.7 | Normalized Weibull Hazard Rate. | 110 |
| 3.8 | Use of the Normalized Weibull Hazard Rate Curves to Obtain Offset Time. | 112 |
| 3.9 | Failure Rates for Selected Values of β , λ , and t_0 which start at the same initial value. | 115 |
| 3.10 | Use of an OR Gate to Obtain a Failure Rate Time Dependence Which Does Not Follow a Power Law. | 117 |
| 4.1 | Sensitivity of Component Unavailability to Demand Verses Standby Failure Mechanisms, Constant Standby Failure Rate, Downtime per Test = 1 Hour. | 120 |
| 4.2 | Sensitivity of Component Unavailability to Demand Verses Standby Failure Mechanisms, Beta = 2 Standby Failure Rate, Downtime per Test = 1 Hour. | 124 |
| 4.3 | Effect of Periodic Testing on a Component With a Constant Standby Failure Rate and Various Magnitudes of Demand Failure Rate. | 127 |
| 4.4 | Graphical Representation of the Time Integrated Unavailability of a Periodically Tested Component. | 131 |
| 4.5 | Comparison of OPTEST and RUN Calculations of the Average Unavailability of a Periodically Tested Component Verses Test Interval. | 140 |

List of Figures

| <u>Figure</u> | <u>Description</u> | |
|---------------|---|-----|
| 4.6 | Average Unavailability Contours for a Periodically Tested Component Having an Effective Downtime per Test of One Hour. | 148 |
| 4.7 | Instantaneous Unavailability Resulting From Hazard Rates Having Three Different Time Dependencies. | 150 |
| 4.8 | Effect of Variations in the Maintenance Interval on a Time Dependent Hazard Rate. | 156 |
| 4.9 | Long Term Effect of Test Caused Wear-out in Standby Failure Mechanisms. Example for $f_{\lambda} = 1.01$. | 159 |
| 5.1 | Cut Set Generator Flow Chart. [Ka80] | 165 |
| 5.2 | Equivalent Transformation of EOR, NAND, NOR, and NOT Gates. [Ka80] | 168 |
| 5.3 | Effect of Demand Failure Mechanisms on the Optimum Test Interval of a Parallel Component System. | 173 |
| 5.4 | Comparison of Testing Policies for Parallel Components with Unequal Unavailabilities During the Test Period. | 175 |
| 5.5 | Single Line Diagram of the 1-out-of-3, Two Valve per Redundancy, Test System Used in the Comparison Between FRANTIC and ICARUS. | 177 |
| 5.6 | Fault Tree of the 1-out-of-3, Two Valve per Redundancy, Test System. | 179 |
| 6.1 | Simplified Diagram of the High Pressure Coolant Injection System of a Boiling Water Reactor. | 191 |
| 6.2 | Optimum Test Interval of HPCI Turbine/Pump as a Function of Composite Failure Rate and Effective Downtime Per Test. | 218 |
| 6.3 | Average Unavailability Contours for HPCI Turbine/Pump Testing When Effective Downtime per Test is One Hour. | 220 |
| 6.4 | Average Unavailability Contours for HPCI Turbine/Pump Testing When Effective Downtime per Test is Two Hours. | 221 |

List of Figures

| <u>Figure</u> | <u>Description</u> | |
|---------------|---|-----|
| 6.5 | Average Unavailability Contours for HPCI Turbine/Pump Testing When Effective Downtime per Test is Three Hours. | 222 |
| 6.6 | Average Unavailability Contours for HPCI Turbine/Pump Testing When Effective Downtime per Test is Four Hours. | 223 |
| 6.7 | Super Component Unavailability Verses Test Interval for the Various Combinations of Demand and Standby Failure Rates Given in Table 6.3. | 224 |
| 6.8 | Effect of Test Caused Wearout on Optimum Test Interval. | 226 |
| 6.9 | Simplified Diagram of HPCI Initiation Function Components. | 228 |
| 6.10 | Initiation Logic Signal Flow. | 229 |
| 6.11 | Average Unavailability of Initiation Sensors Without Common Cause Failures. | 233 |
| 6.12 | Average Unavailability of Initiation Sensors With Common Cause Failures Within Individual Groups of Sensors. | 235 |
| 6.13 | Average Unavailability Due to Online Testing of the Initiation Logic Relays. | 240 |
| 6.14 | Unavailability of Initiation Logic Relays. With Current Design and Current Test Procedures and Staggering. | 243 |
| 6.15 | Wiring Diagram of HPCI Initiation Logic with Modifications 2 and 3. | 246 |
| 6.16 | Changes in Relay Logic Signal Flow Produced by Modification Two. | 247 |
| 6.17 | Effects of Design Modifications on Initiation Function Unavailability Verses Periodic Test Interval. Effective Downtime per Test = 2.0 hours. | 249 |
| 6.18 | Sensitivity of Modified Initiation Logic Circuit Unavailability to an Order of Magnitude Increase in Component Failure Rates. | 251 |

List of Figures

| <u>Figure</u> | <u>Description</u> | |
|---------------|---|-----|
| 6.19 | Autoisolation Function Unavailability as a Function of Autoisolation Sensor Test Intervals. | 258 |
| 6.20 | Injection Function Unavailability Due to Autoisolation Sensor Tests Which Cycle Autoisolation Valves MO 2301-4 and MO 2301-5. | 264 |

List of Tables

| <u>Table</u> | <u>Description</u> | |
|--------------|---|-----|
| 2.1 | Table 4 of ANSI/IEEE Standard 352-1975. Test Interval as a Function of Logic Configuration and Unavailability Design Goal G. | 28 |
| 2.2 | Component Unavailability Equations for Four Classes of Components. [Ca77] | 37 |
| 3.1 | Parameters Used for Plotting Curves in Figure 3.5. | 103 |
| 3.2 | Values of β , λ , and t Used to Obtain the Hazard Rate Curves Shown in Figure 3.6. | 107 |
| 4.1 | Component Failure Parameters for Figures 4.1 and 4.2 and Plots of the Resultant Hazard Rates as a Function of Time. | 122 |
| 4.2 | Input and Output of OPTTEST Calculations for Figure 4.5. | 141 |
| 4.3 | Input and Output of RUN Calculations for Figure 4.5. | 143 |
| 4.4 | Comparison of Calculations Using the Old-Old Renewal Option and a $\beta=3$ Hazard Rate With Those Using an Equivalent Constant Failure Rate. | 154 |
| 5.1 | Input and Output of CUTSETS Application to the Test System. | 180 |
| 5.2 | Input Parameters for the 1-out-of-3, Two Valves Per Redundancy, Test System. | 181 |
| 5.3 | Input of FRANTIC II-MIT Application to the Test System. | 181 |
| 5.4 | Results of Comparison of Uncorrected and Corrected Versions of FRANTIC with Vaurio's Calculations. | 182 |
| 6.1 | HPCI Injection Function Single Component Cut Sets. | 202 |
| 6.2 | Assessed Range of Single Component Cut Sets Tested by the HPCI Turbine/Pump Test. | 211 |
| 6.3 | Range of Super Component Failure Rates for Turbine/Pump Test. | 212 |

List of Tables

| <u>Table</u> | <u>Description</u> | |
|--------------|---|-----|
| 6.4 | Upper Bound Estimates of MO 2301-3 Standby Failure Rate Since 1975. | 213 |
| 6.5 | Probability of Test Caused Failure to Injection Function Failure Events 31 and 32 (Isolation Valves, NOFC) as a Result of Autoisolation Function Tests. | 263 |
| 6.6 | Summary of Periodic Test Recommendations. | 267 |

CHAPTER 1

INTRODUCTION

Standby safety systems have the very difficult mission of remaining idle for long periods of time while being prepared to function under accident conditions at a moment's notice. The operational status of most of these systems can not be monitored while they are idle, so the Nuclear Regulatory Commission (NRC) requires that systems important to safety "be designed to permit appropriate periodic inspection of important areas and features..." [10CFR50, App. A] Unfortunately, establishing a quantitative basis for judging what is appropriate is very difficult for a complex safety system containing many components. As a result, periodic test and inspection policies are frequently based on "engineering judgment" or the analysis of equivalent single component systems, rather than a quantitative balancing of the advantages and disadvantages of accomplishing a particular testing program in the context of the entire system's safety function.

To aid in establishing a more quantitative basis for periodic testing, the NRC has developed and distributed the FRANTIC (Formal Reliability Analysis including Normal Testing, Inspection and Checking) computer codes. [Ve77, Ve81] The codes use time dependent unavailability analy-

sis to accomplish this task. Given a comprehensive set of input parameters describing component failure rates and periodic testing policies and a user supplied equation relating the system's unavailability to that of the components, they calculate the system's instantaneous unavailability at all important time points and the average system unavailability over a user specified calculation period. Two versions are currently available. The original FRANTIC code assumes constant component failure rates, while FRANTIC II can model wear-out and burn-in as a function of both calendar time and periodic tests. While the codes have been applied to illustrative examples [eg. EP1443, Va79b, Ka80], to the best of this author's knowledge, they have not yet been used for a detailed examination of the periodic testing program for an operating reactor system.

The purpose of this thesis is to assess the utility of time dependent unavailability analysis for improving the availability of standby safety systems using the FRANTIC II computer code as a tool. To accomplish this, FRANTIC II is assessed from an engineering point of view and modified as necessary to make it more useful for applications to operational systems. It is then interfaced with a cutset generator and evaluator so that it can be applied to complex system models. The modified version of the code is named FRANTIC II-MIT. The code is then applied to the High Pressure Coolant Injection System of a Boiling Water Reactor,

and a quantitatively based periodic testing policy keyed to a fault tree evaluation of the system's safety functions is established. As a result, this thesis provides an improved framework within which a systems engineer can establish a quantitative basis for a periodic testing program.

1.1 BACKGROUND AND MOTIVATION

To illustrate the primary motivation for accomplishing periodic testing, consider the following simple example. Figure 1.1 represents the time dependent unavailability of a standby safety system whose failure rate can be modeled by a single constant standby failure rate, λ_s per hour, and which can be tested in its entirety at one time. This parameter models the system's susceptibility to random shocks that transfer it into a state which can not respond to a demand to operate. [Ba75, Ap76, EP1443] However, because the system is idle, the shocks do not produce observable effects until the demand is actually made. As the system sits idle, the time during which the shocks can occur lengthens and the probability that a failure has occurred gradually increases. When a test is accomplished, the demand to operate is made and the failures are revealed and immediately repaired. After the test the system's unavailability is zero because random shocks have not yet had an opportunity to occur.

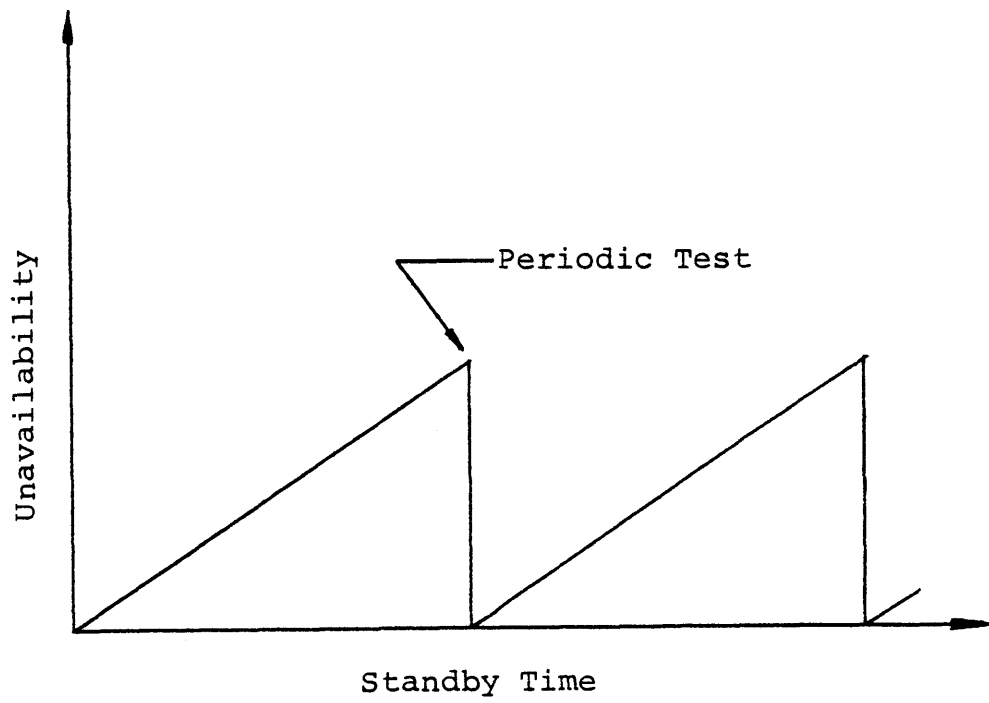


Figure 1.1. Simplified Example of a Periodically Tested System

In practice, standby failures are not the only factor to consider in establishing a periodic test policy. For example:

- Shocks may occur during the demand as well as during the standby period. They produce failures whose probability is independent of the standby period. If demand related failures are possible, the unavailability of a component is not reduced to zero by an operational test.
- Frequently a system must be reconfigured to test an accident mitigation function without interfering with normal operations, and it may be unavailable to perform that function in the event that a true demand occurs during the test.
- Operational tests which cycle the system to an active mode may cause wear-out that makes the system more susceptible to failures later in its life.
- Since test conditions are not always similar to accident conditions, a periodic test may not be able to detect all the failure mechanisms which could prevent the system from performing its intended function.
- The act of accomplishing the test can cause failures which require repair and thus produce additional unavailability.
- Human error during the test may leave the system in a failed state at its conclusion.

Clearly, there are positive and negative aspects of periodic testing which must be balanced when formulating a periodic testing program.

Even if one accounts for all the factors mentioned in the previous paragraph, a simple one component system frequently can not be used to model operational systems. Engineered safeguards systems in a nuclear reactor are a prime example. They contain many components which exhibit a variety of failure mechanisms. Within the context of these complex systems, all the considerations listed for the simple system of the previous paragraph now apply to each component. It is difficult to establish an optimal testing policy for these systems for several reasons:

- Direct testing of the entire safety function usually can not be performed without interfering with the operation of the reactor, so portions of the system must be reconfigured, disabled or bypassed, while other parts come into closer alignment with their operational configuration
- The system is frequently designed to respond to diverse indications of an accident, each of which must be tested separately.
- The act of testing some components can affect the status of both the component being tested and other groups of components.

- Testing of individual components can cause failures which must be repaired. Frequently the entire system is disabled during the repair.

Until recently, a reasonable tool for establishing a quantitative basis for a periodic testing program which considers all of the above factors did not exist. It is very difficult to evaluate the effect of individual component failure mechanisms on the functioning of a larger system having many different types of components without a tool such as a computer code to reduce the computational difficulties. The Nuclear Regulatory Commission has recently developed the FRANTIC II computer code to alleviate this problem. The code calculates component unavailability at specific points in time taking into account both demand and time dependent failure rates and modeling both the positive and negative effects of periodic testing. A user provided equation is then called to calculate the system unavailability in terms of individual component unavailabilities. Because it has not yet been applied to actual system problems, FRANTIC II's capabilities and limitations have not yet been explored.

1.2 PURPOSE AND SCOPE

The purpose of this thesis is to assess the usefulness of time dependent unavailability analysis for improving the availability of standby safety systems using the FRANTIC II

computer code as a tool. To accomplish this goal, the following tasks are performed:

1) Modification of FRANTIC II as necessary to provide modeling capability of physically reasonable failure mechanisms. The resulting code is named FRANTIC II-MIT. The task includes:

- An engineering interpretation of failure mechanisms of standby components subject to periodic testing and repair.
- Correlation of FRANTIC II input parameters with these mechanisms
- Incorporation of an additional model which accounts for test caused changes in the demand failure rate.
- Incorporation of an offset time into the Weibull hazard rate to make possible the modeling of a family of time varying failure rates having any initial or final value.
- Provision for human error during periodic testing which results in the nondetection of a fraction of the failures which the test is capable of detecting.

2) Modifications to improve the code's capability to examine the sensitivity of component and system unavailability to input parameter changes. This includes:

- Interface of the code with a cutset generator with provisions to save the cutsets for reuse as required.
- Addition of subroutine OPTTEST, which can calculate the optimum test interval of a component for a given set of

input parameters, assuming all parameters have constant failure rates.

3) Investigation of the importance of some of the code's modeling capabilities relative to assumptions commonly used in practical unavailability analysis. This includes:

- Demand failures versus standby failures.
- Effects of the various types of imperfect testing.
- Effects of calendar time dependent failure rates (commonly called wear-out and burn-in).
- Effects of test dependent failure rates (test caused wear-out or the effects of product improvement due to elimination of failure causes).

4) Application of FRANTIC II-MIT to the High Pressure Coolant Injection System of a Boiling Water Reactor to obtain an understanding of the factors which can influence the selection of periodic testing intervals. The analysis includes:

- Description of the system with particular attention to the interaction of component testing policies within the system and types of component failures mechanisms.
- Construction of fault trees down to the smallest testable component level.
- Quantification of the fault tree using generic data and, to the extent that it is available, plant specific data.

- Analysis of periodic test procedures to estimate quantitative test input parameters.
- Sensitivity studies of a number of testing options to determine the most important contributors to safety function unavailability and the effect that the testing policy can have on these contributors.
- Discussion of the practical problems involved in addressing real systems problems using time dependent unavailability analysis with recommendations for solutions and/or further research.

This thesis provides an improved framework within which a systems engineer can establish a quantitative basis for a periodic testing program. The observations and recommendations resulting from the application of time dependent unavailability analysis should lead to a more rational testing program and an improvement in the performance of those systems.

1.3 STRUCTURE OF THESIS

Chapter 2 reviews the basic concepts of unavailability analysis and summarizes regulations and research which address periodic testing of standby components.

Chapter 3 presents an engineering interpretation of the FRANTIC II code and the version of it developed by this work, FRANTIC II-MIT. It provides explanations and examples to assist the systems engineer in identifying what each

input parameter can model. It outlines the modifications incorporated in FRANTIC II-MIT and provides the guidance necessary for its use. An appendix outlines the code's input format.

Chapter 4 uses FRANTIC II-MIT to investigate the unavailability of single component systems. The practical implications of various assumptions about the failure mechanisms of components are illustrated through examples. Where possible the code's calculations are compared with analytical expressions derived in the literature. The subroutine OPTTEST is presented in this chapter. It was designed to quickly calculate the optimum test interval of single components having a constant failure rate.

Chapter 5 describes the use of FRANTIC II-MIT in conjunction with the cut set generation and evaluation subroutines of UNRAC [Ka80] and examines its applications to a few simple component configurations. It then describes techniques for applying the package to more complex systems, using the calculations presented in Chapter 6 as the primary example. An appendix summarizes the input format necessary to use the cut set generation and evaluation subroutines, which have been assembled into a code called CUTSETS. Another appendix presents IBM CMS/VS system specific programs which can tailor the code's input and output files to suit the needs of the user.

Chapter 6 uses the FRANTIC II-MIT/CUTSETS package to examine the periodic testing policy of a Boiling Water Reactor's High Pressure Coolant Injection System in detail. This chapter demonstrates that the package can be a powerful and versatile tool for examining the consistency of periodic testing policies.

Chapter 7 summarizes the results of this study and makes recommendations for future research.

CHAPTER 2

UNAVAILABILITY AND PERIODIC TESTING

This chapter presents a basic description of the unavailability analysis of standby safety systems. First unavailability is defined in relation to the operational requirements of a standby safety system. Then basic concepts of unavailability analysis are described as they apply to monitored and periodically tested components. Since an engineering interpretation of each failure and test parameter is presented in Chapter 3, this section focuses on those points necessary to explain what the status of a failed component can be and how long that status can last. Finally, a review of regulations and research addressing periodic testing of standby systems using unavailability analysis is presented.

2.1 STANDBY SYSTEM UNRELIABILITY DEFINITIONS

In reactor safety applications, fault tree analysis is commonly used to account for the ways that the failure modes of components composing the system contribute to system failure. A number of excellent references [He81, NUREG-0492, McC81] discuss the construction and use of fault trees. In this thesis it is assumed that the reader is familiar with these techniques.

Fault trees are generally used to calculate system unavailability and cumulative failure probability. These two quantities are used to describe the likelihood that the system will not complete its required mission. This study addresses the calculation of unavailability. However, so that system unavailability can be put into proper context, it is useful to define these terms in the context of the mission of a standby safety system before proceeding further.

Unavailability, $Q(t_1)$

The ability of a standby system to start when required depends on its being in state which is capable of making the transition from the idle mode to the active mode at the time of the demand. The probability of not being in such a state at a point in time, t_1 , is referred to as the system's unavailability, $Q_s(t_1)$, to allow the transition.

The unavailability of a complex system will depend on individual component unavailabilities, $q_i(t_1)$, and the combination of component faults required to produce system failure, as represented by a fault tree. The top event of the fault tree is the failure of the system to startup to the fully operational properly aligned active mode, given a demand of a specified type. The unavailability of a component is the probability that it can not perform at time t_1 the specific functions required of it by the system to startup.

The symbol $Q_i(t_1)$ is defined to be the unavailability of minimal cut set i at time t_1 . In fault tree analysis a cut set defines a combination of component failures whose simultaneous existence is both necessary and sufficient to produce system failure. A minimal cut set is one which is not a subset of another cut set, and its unavailability is the product of the unavailabilities of all the components in it. A large number of minimal cut sets are obtained from most system fault trees, and the upper bound of the system unavailability is $Q_s \leq \sum Q_i$. Knowledge of the individual values of Q_i is useful for determining which combinations of component failures are most likely to cause system failure.

The unavailability to make a transition to the active mode can be a result of failures that occur either during the standby period prior to the demand or during the actual transition. Failures which occur before the demand may be detected and repaired. If the repair is completed before the demand is made, the system will be available. It is not necessary that the system remain in an operable state for the entire standby interval so long as it is operable when the demand to transfer to the active mode is made.

Cumulative Failure Probability, $P(t_1, t_2)$

Given that a system has successfully started to perform its function during an accident, the probability that it will not continue to perform its function successfully for the entire mission interval time (t_1, t_2) is called the cumu-

lative failure probability, $P(t_1, t_2)$, of the system. The cumulative failure probability is calculated using a fault tree having a top event which is the failure to remain properly aligned and successfully performing its function under specified performance criteria.

A typical mission requirement is that the system perform its function for the entire period from t_1 to t_2 . This requirement is specified when adverse effects will result immediately from a stoppage of the safety function.

However, some systems may be allowed to be down for repair for short periods without resulting in adverse consequences. An example would be those systems which provide long term removal of decay heat from a reactor. After the first few days mission failure could be defined as the system being down for more than a specified interval. The allowed downtime could be extended as the time since shutdown increases.

Like the startup unavailability, the cumulative failure probability depends on the structure of the system and the failure characteristics of its components. However, the fault tree whose top event defines a failure to make a transition upon demand may be quite different from one which describes the failure to continue running once the transition has been successfully accomplished. For example, the transition might require that valves change position. Once they have done so, they must remain passive for the duration

of the mission. Therefore, a failure mode of the valve for transition unavailability would be failure to make the required position change, whereas a failure mode for the cumulative failure probability during the active phase would be a change from the proper alignment to a position which would prevent the proper functioning of the system.

Although the fault trees quantifying unavailability and cumulative failure probability are different, some component failure modes may be the same in both. For example, emergency core cooling systems are designed so that an isolation signal is generated if sensors indicate that the source of a loss of coolant accident comes from within that system. The isolation signal will either prevent the system from starting or will shut it down if it is already running. Therefore, the production of an erroneous isolation signal would appear as a failure mode in both fault trees.

Unreliability, $F(t_1, t_2)$

The probability that a standby system will either not start or not run for the required mission time is called in this thesis the unreliability, $F(t_1, t_2)$, of the system. Other terms are system undependability and system failure probability. System unreliability may be expressed as the sum of the unavailability and cumulative failure probability as follows:

$$F(t_1, t_2) = Q_s(t_1) + \{1 - Q_s(t_1)\}P(t_1, t_2) \quad (2.1)$$

This thesis concentrates on the modeling of a system's unavailability to initiate a safety function and not on the cumulative failure probability. More specifically, it focuses on its time dependence and the effects that periodic testing might have on both the instantaneous and the time averaged unavailability of a safety system.

2.2 BASIC UNAVAILABILITY CONCEPTS

Probabilistic unavailability analysis requires determining when component failures occur and for how long they last. For this purpose components can be divided into two generic groups, periodically tested and "other."

2.2.1 "OTHER" COMPONENTS

Figure 2.1 is a graphical representation of the asymptotic unavailability of all types of components other than periodically tested. There are three contributors to the component's unavailability:

1) Standby failures which occur at a constant rate of λ per hour and are detected and repaired with an average downtime of T_R . (More commonly known as monitored failures.) These failure occur to components which perform some type of function during standby, so that failures can be identified when they happen. For illustrative purposes,¹ the steady state

¹ For a more complete treatment of the probabilistic parameters of components with binary states see, for example, Henley and Kumamoto. [He81]

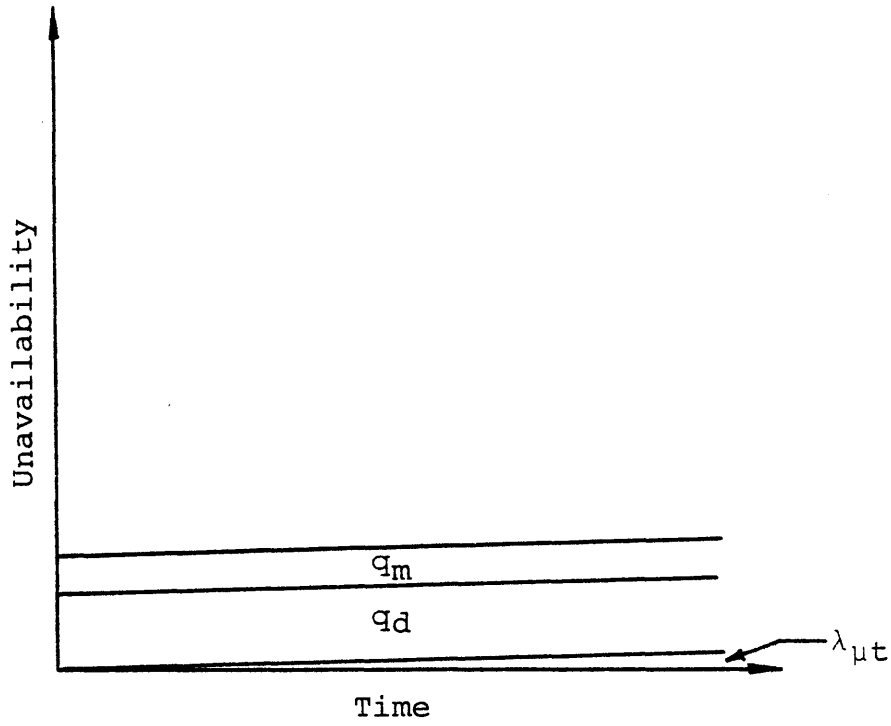


Figure 2.1. Time Dependent Unavailability of Components Other Than Those Which Must Be Periodically Tested to Reveal Standby Failures.

unavailability due to the downtime, T_R , during which the failures are detected and repaired can be heuristically derived by recognizing that, in some increment of time, dt , the unavailability of a component is increased by the probability that it fails in dt and decreased by the probability that it is repaired in dt . At steady state $q_m(t)$ is a constant, q_m , and $d[q_m(t)] = 0$. Therefore:

$$[1 - q_m] \lambda dt - q_m (1/T_R) dt = 0 \quad (2.2)$$

Where:

λ - Conditional failure rate (assumed constant)

$1/T_R$ - Rate at which repairs are completed (assumed to behave as an exponential process)

Rearranging,

$$q_m = \frac{\lambda T_R}{1 + \lambda T_R} \approx \lambda T_R \text{ when } \lambda T_R < 0.1 \quad (2.3)$$

2) Transition failures, modeled by a constant time independent unavailability per demand, q_d . These failures occur because of a change in the component's operating conditions at the time of the accident, including the possibility of operator error.

3) Failures which occur at a rate of λ_μ per hour during the standby period, but for some reason are not detected until

the component fails to operate under the conditions of the true demand. The rate at which the component's unavailability due to these failures increases can be expressed as follows:

$$d[q(t)] = [1-q(t)]\lambda_{\mu}dt \quad (2.4)$$

Where:

$\lambda_{\mu} dt \equiv$ Conditional probability that the component fails between t and $t+dt$, given that it is working at t .

$[1-q(t)] \equiv$ Probability that the component is working at time t .

In its most general form λ_{μ} can be a function of time. (Chapter 3 shows how to model time dependent failure rates with a generalized Weibull hazard rate.) For convenience it is assumed here that λ_{μ} has a constant value.

The equation can be rearranged to:

$$\frac{d[q(t)]}{[1-q(t)]} = \lambda_{\mu}dt \quad (2.5)$$

and integrated from time 0 to t , yielding

$$-\ln[1-q_{\mu}(t)] + \ln[1-q_{\mu}(0)] = \lambda_{\mu}t \quad (2.6)$$

Since it is assumed that the component was working at $t = 0$, $q_{\mu}(0) = 0$, giving $\ln(1) = 0$. This leads to:

$$q_{\mu}(t) = 1 - e^{-\lambda_{\mu}t} \approx \lambda_{\mu}t \quad \text{for } \lambda_{\mu}t < 0.1. \quad (2.7)$$

The three failure modes are accounted for together because they all behave as a function of just the standby time. This is not true with periodically tested components, which will be discussed in the next section.

2.2.2 PERIODICALLY TESTED COMPONENTS

There are three major differences between periodically tested components and other types of components:

1) Although λ failures occur randomly in time, they are not detected or repaired randomly in periodically tested components. Repair can not be started until the failure is detected, and failures can not be detected until the component is tested. Thus detection and repair occur at definite points in time which are controlled by the periodic testing policy.

2) Because periodic testing generally requires that the component be cycled to its active mode to detect failures, the act of testing can cause additional failures which contribute to the component's unavailability.

3) Whereas T_R , the average downtime before a monitored component can be restored to an operational condition, was a major contributor to the unavailability of monitored components, the repair time of periodically tested components is a relatively minor contributor compared to the time during which the failure can have occurred but be undetected because a test has not yet revealed it.

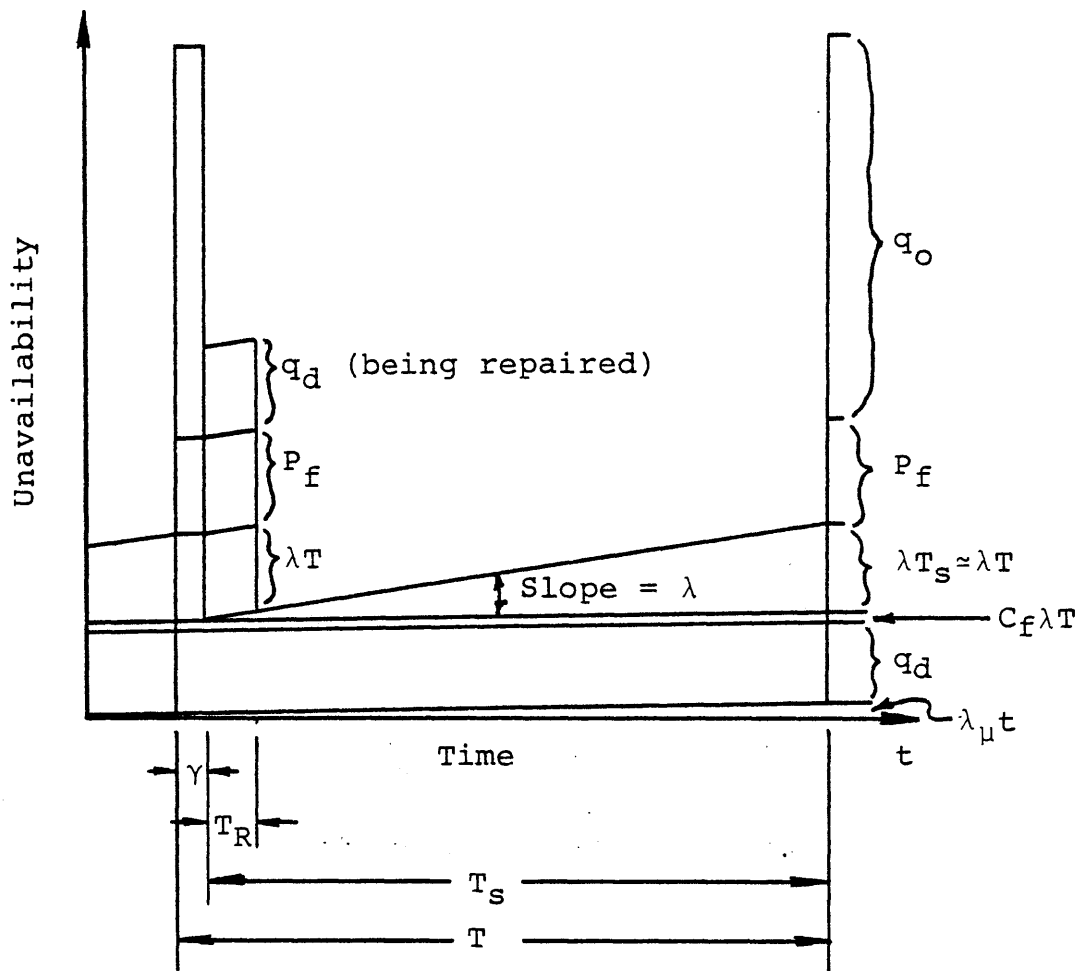


Figure 2.2. Time Dependent Unavailability of Components Which Must Be Tested to Reveal Standby Failures.

Figure 2.2 shows the time dependence of a periodically tested component. Instead of only one, there are now three distinct time frames for which component unavailability must be determined:

τ \equiv Test period. During this time the component is cycled to its active configuration to verify its operability. If it is found failed during this time it is assumed to remain failed for the entire test period.

T_R \equiv Repair Period. Failed components are assumed to remain failed until repair is completed, which takes an average of T_R time. If components are verified operational by the test, they go back on standby during this time.

T_S \equiv Standby period. For most practical applications, $T_S \gg (\tau + T_R)$, so $T_S \approx T$, the interval between the beginning of consecutive periodic tests.²

During standby a periodically tested component can be made unavailable for the same reasons as the other types of components. However, since it is usually idle during standby, failures will not be revealed until the component is required to perform its function. Assuming that standby failures occur with a constant conditional failure rate of λ per hour, the probability that they have occurred increases in exactly the same manner as that of undetectable failures modeled by λ_μ in Equations (2.4) to (2.7). However, now the

² In FRANTIC II, the test interval is given the symbol T_2 .

effective time period starts at t_w , the last time the component was known to be working. The resulting unavailability is:

$$q_{\lambda}(t_w, t) = 1 - e^{-\lambda(t-t_w)} \approx \lambda(t-t_w) \quad (2.8)$$

When the component is tested, detectable standby failures are revealed, but other factors also influence the components unavailability.. The various contributions to a periodically tested component's unavailability during the test and repair periods are:

q_d - Demand failures

$q_u(t)$ - Probability of undetectable standby failures.

This continues to rise throughout the component's life independent of standby, test, and repair (unless renewal occurs).

$q_{\lambda}(t_w, t_w+T)$ - Probability that a detectable standby failure exists at the beginning of a periodic test following a standby interval of T. ($\approx \lambda T$ if < 0.1)

P_f - Probability that the test causes failures which require repairs.

q_o - Probability that a component can not respond to a true demand while it is being tested.

C_f - Probability that detectable standby failures are not detected at a periodic test due to human error.

2.3 CURRENT STATUS OF ANALYSIS OF PERIODIC TESTING

2.3.1 REGULATIONS AND STANDARDS

Requirements for periodic testing of standby safety systems are currently set forth in 10 Code of Federal Regulations, Part 50, and ANSI/IEEE Std 338-1977, Criteria for the Periodic Testing of Nuclear Power Generating Safety Systems.

Periodic testing is specifically required in a number of the design criteria set forth in Appendix A to 10 CFR 50. However, the extent of testing necessary to satisfy the criteria is not specified. Instead, general phrases are used. For example, Criterion 18 - "Inspection and testing of electric power systems," states:

Electric power systems important to safety shall be designed to permit appropriate periodic inspection of important areas and features ... to assess the continuity of the systems and the conditions of their components.

More specific requirements and criteria for periodic testing are set forth in ANSI/IEEE Std 338-1977. It provides guidance for the development of procedures and documentation, and the design of equipment necessary for the periodic testing of a nuclear power generating station's protection and power systems. This standard provides an outline of good engineering practice and records requirements to be used in accomplishing and documenting the tests. The question of test interval is addressed in Appendix A1, which states:

Determination of test intervals based on mathematical relations involving logic,

failure rate data, test duration, and permissible system unavailability is covered by IEEE Std 352-1975, Guide for General Principles of Reliability Analysis of Nuclear Power Generating Station Protection Systems.

Section 7 of ANSI/IEEE Std 352-1975 provides the guidance for the establishment of test intervals. For a single component system the test interval is expressed as:

$$\theta = 2G/\lambda \quad (2.9)$$

where:

G \equiv Unavailability design goal

λ \equiv Standby failure rate

θ \equiv Periodic test interval

The test intervals for systems of components arranged in common logic configurations are given in Table 4 of the standard, which is reproduced as Table 2.1.

Equation (2.9) is essentially a rearrangement of the expression for the approximate average unavailability due to detectable standby failures which have occurred, but have not yet been revealed by a periodic test. The average unavailability can be easily found from Figure 2.2. The triangular area represents the increasing probability of unrevealed failures. The average value is half of the maximum, or $\theta\lambda/2$ by the current notation.

The standard does not quantitatively account for downtime unavailability during the test period or the effects of imperfect testing, although it does mention some of the prob-

**Test Interval as a Function of Logic Configuration
and Unavailability Design Goal G**

| Logic Configuration | Test Interval | |
|---------------------|--|---|
| | Simultaneous Testing | Perfectly Staggered Testing |
| 1/2 | $\frac{1}{\lambda} \times (3\bar{G})^{1/2}$ | $\frac{1}{\lambda} \times \left(\frac{24\bar{G}}{5}\right)^{1/2}$ |
| 2/2 | $\frac{1}{\lambda} \times \bar{G}$ | $\frac{1}{\lambda} \times \bar{G}$ |
| 1/3 | $\frac{1}{\lambda} \times (4\bar{G})^{1/3}$ | $\frac{1}{\lambda} \times (12\bar{G})^{1/3}$ |
| 2/3 | $\frac{1}{\lambda} \times (\bar{G})^{1/2}$ | $\frac{1}{\lambda} \times \left(\frac{3\bar{G}}{2}\right)^{1/2}$ |
| 3/3 | $\frac{1}{\lambda} \times \frac{2\bar{G}}{3}$ | $\frac{1}{\lambda} \times \frac{2\bar{G}}{3}$ |
| 1/4 | $\frac{1}{\lambda} \times (5\bar{G})^{1/4}$ | $\frac{1}{\lambda} \times \left(\frac{7680\bar{G}}{251}\right)^{1/4}$ |
| 2/4 | $\frac{1}{\lambda} \times (\bar{G})^{1/3}$ | $\frac{1}{\lambda} \times \left(\frac{8\bar{G}}{3}\right)^{1/3}$ |
| 3/4 | $\frac{1}{\lambda} \times \left(\frac{\bar{G}}{2}\right)^{1/2}$ | $\frac{1}{\lambda} \times \left(\frac{8\bar{G}}{11}\right)^{1/2}$ |
| (1/2) x 2 | $\frac{1}{\lambda} \times \left(\frac{3\bar{G}}{2}\right)^{1/2}$ | $\frac{1}{\lambda} \times \left(\frac{12\bar{G}}{5}\right)^{1/2}$ |

Table 2.1. Table 4 of ANSI/IEEE Standard 352-1975
Test Interval as a Function of Logic Configuration
and Unavailability Design Goal G

lems and tradeoffs the analyst should address. Therefore it should not be applied without supplemental quantitative analysis.

2.3.2 PUBLISHED RESEARCH

Many of the tradeoffs to be considered when establishing a periodic test and maintenance policy for a standby safety system have been addressed in the literature. However, only specific parts of the problem have been addressed in any one paper and application has been restricted primarily to simple one component systems whose failure rate can be represented by a single distribution, or combinations of components in standard logic configurations, such as 2-out-of-3:Good.³

Simple Systems With Test Downtime

Jacobs [Ja68] and Epstein and Shiff [Ep68] were the first to consider the periodic testing of components which are made unavailable to accomplish their safety function while being tested. They suggested that an optimum test interval could be derived for this type of system. Figure 2.3 illustrates this concept. It shows the unavailability of a single component system for three different periodic test intervals. At the end of each test the system is known to be working, so the unavailability is zero. The failure rate of the component is assumed constant during standby, and the unavailability rises exponen-

³ The term 2-out-of-3:Good means the system works if 2 of its 3 components are working.

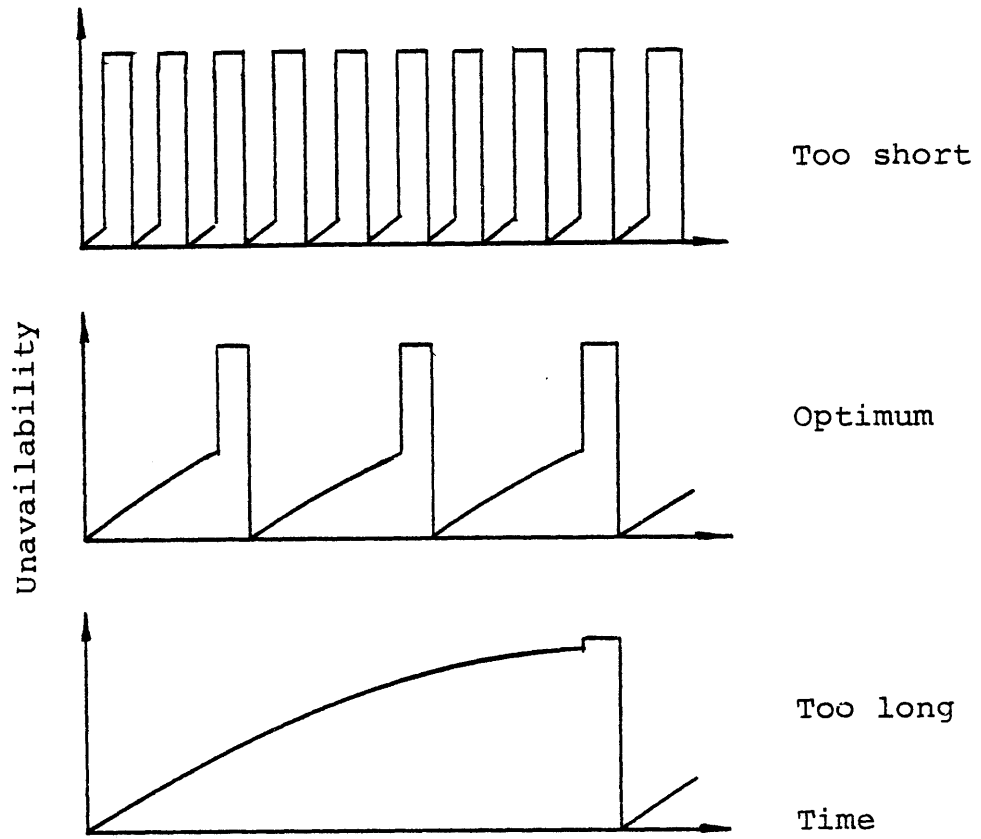


Figure 2.3. Example of the Effect of Test Interval on Unavailability.

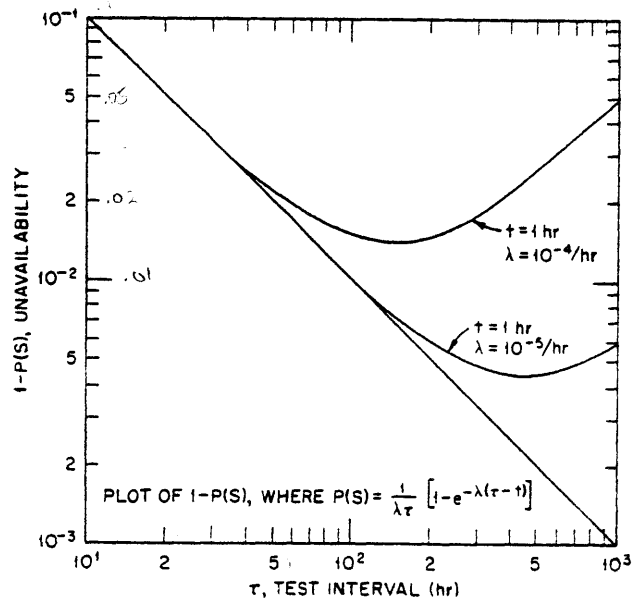


Figure 2.4. Unavailability of a Component Subject to Standby Failures and Test Downtime. [Ja68]

tially until the next test is accomplished. Since the safety function of the component is assumed to be bypassed to accomplish this test, its unavailability rises to one. If the test interval is very short, the component would be bypassed most of time for testing and would have high unavailability. Conversely, if the component is tested with an extremely long interval, its unavailability approaches one and remains there for a very long time. This suggests that there might be a test interval for a given system failure rate and test down time that minimizes the unavailability.

Using this concept, both authors derive an equation which expresses the average unavailability of a one component system as:

$$Q(S) = 1 - \frac{1}{\lambda\tau} [1 - e^{-\lambda(\tau-t)}] \quad (2.10)$$

Where,

λ \equiv Conditional failure rate of the system (per hour)

τ \equiv Periodic test interval (hours). [Equivalent to T_2
(which has units of days) in FRANTIC]

t \equiv Total time that the system is down per cycle due to
testing (hours). [Equivalent to $(q_0\tau)$ in FRANTIC]

A plot of this equation taken from Jacobs' paper is shown in Figure 2.4. (Note: Jacobs originally wrote an equation for availability, which he called $P(S)$, but he plotted curves for unavailability, which show the results better. The unavail-

ability equation is written here for consistency with the remainder of the thesis.) The curves in Figure 2.4 are plotted for a test downtime of 1 hour and two different failure rates. The curves dip through a minimum, indicating that there is an optimum test interval for a system to be taken out of service for testing. As the failure rate increases, the optimum test interval decreases.

By differentiating equation (2.10), both authors obtain a simple expression for the optimum test interval in cases which $1/\lambda \ll t$ and $\lambda\tau < 0.1$:

$$\tau = \frac{2t}{\lambda} \quad (2.11)$$

Note that this equation can be rearranged so that

$$t(1) = \lambda\tau \quad (2.11a)$$

At the optimum test interval the area under the triangle in Figure 2.3 (representing the contribution of standby failures which have occurred but have not been detected) equals the area of the rectangle (downtime contribution of the test which detects them).

A recent Nuclear Regulatory Commission document, NUREG/CR-2158, duplicates and expands upon the early work described above. The report states explicitly the following

assumptions in deriving the optimal test intervals: (These same assumptions were implicit in earlier papers.)

1. The component has a constant standby failure rate of λ per hour.
2. Testing is done periodically and is done on line, i.e., during the test the component could be called upon to operate.
3. During the time of the test, the component is unavailable and unable to respond if called upon to operate.
4. The testing requires an average time period τ to complete.
5. Other than the test time τ during which the component is unavailable there are no test-caused failures or degradations such as those due to human errors.

The equations derived in NUREG/CR-2158 are the same as those derived by Jacobs, with the exception that the approximations for $\lambda\tau$ and τ used to derive Equation (2.11) are also applied to the unavailability equation and a slightly different notation is used. The resulting equations are:

$$q_c = \frac{1}{2}\lambda T + \frac{\tau}{T} \quad (2.12)$$

and

$$T_o = \frac{2\tau}{\lambda} \quad (2.13)$$

where:

$T_o \equiv$ Optimum test interval

q_c \equiv Average component unavailability

λ \equiv Component failure rate

T \equiv Constant standby time between tests

τ \equiv Constant test downtime during which the component is
unavailable

(With this notation, a test interval has a duration of $T+\tau$ hours. The test interval was τ in Jacobs' paper and is given the symbol T_2 in FRANTIC.)

Using these equations the report presents a compilation of figures and tables which present optimum test intervals for a variety of component failure rates and test down times selected to cover the range of values normally encountered in nuclear plant operations. For comparison with Jacob's paper, Figure 4 of NUREG/CR-2158 is reproduced here as Figure 2.5. Note that the curves for $\lambda=1.E-4$ and $1.E-5$ are the same as those found in Figure 2.4. The document's tables and graphs constitute a comprehensive application of Jacobs' work and provide a convenient reference for an engineer making a first estimate of test intervals.

Caldarola [Ca77] has derived a set of time dependent unavailability and failure intensity equations for components having constant failure and repair rates. He considers four classes of components, each having a well defined repair policy:

Class 1 Unrepairable components. No repair action is foreseen.

AVERAGE COMPONENT UNAVAILABILITY VERSUS TIME BETWEEN TESTS
 PARAMETRIC WITH COMPONENT FAILURE RATE, λ (FAILURES/HR)

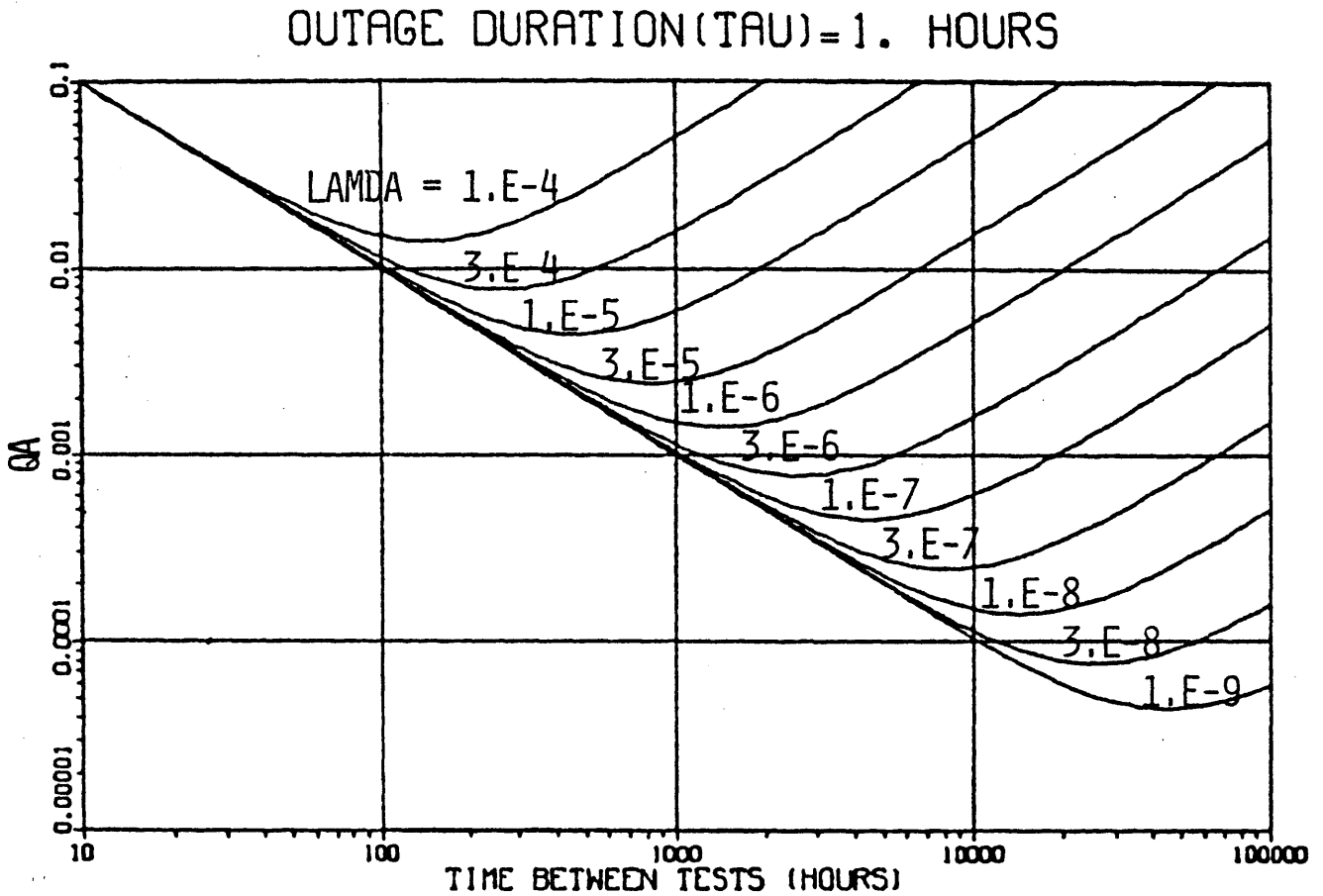


Figure 2.5. Average Component Unavailability Verses Time Between Tests, Parametric With Component Failure Rate, λ (failures/hr), Outage Duration (τ) = 1 hour. (Figure 4 of NUREG/CR-2158).

Class 2 Repairable components with failures which are immediately detected (revealed faults).

Class 3 Repairable components with failures which are detected upon demand (faults remain unrevealed until the next demand occurs).

Class 4 Repairable components with failures which are detected upon inspection.

His results are given in Table 2.1, where:

V_u \equiv Component unavailability with initial state intact

V_d \equiv Component unavailability with initial state failed

λ \equiv Component failure rate (constant)

μ \equiv Component repair rate (constant)

ν \equiv Average demand frequency (constant)

θ \equiv Time needed to inspect an unfailed component

τ \equiv Time needed to repair a failed component during inspection

η \equiv Time interval between two successive inspections

The author states that the motivation for his work was to derive a set of comprehensive and consistent set of equations that address repair and inspection which are usually met in practice for application fault tree analysis. Although the resulting equations are time dependent, they do not appear to have the capability of modeling contributions of imperfect testing to unavailability.

Redundant Standby Components

Hirsch [Hi71] presents nomographs for determining period-

Table 2.2. Component Unavailability Equations for Four Classes of Components. [Ca77]

| Class | Type of Component | Initial State | | |
|-------|--|---|---|---|
| | | Intact (V_u) | Failed (V_d) | |
| 1 | Unrepairable | $1 - e^{-\lambda t}$ | | |
| 2 | Repairable (revealed faults) | $\frac{\lambda}{\lambda + \mu} [1 - e^{-(\lambda + \mu)t}]$ | $\frac{\lambda}{\lambda + \mu} + \frac{\mu}{\lambda + \mu} e^{-(\lambda + \mu)t}$ | |
| 3 | Repairable (faults detected upon demand) $\epsilon = \left(\frac{\lambda + \mu + \nu}{2}\right)^2 - (\lambda\mu + \lambda\nu + \mu\nu)$ | $\epsilon < 0$ | $\frac{\lambda(\mu + \nu)}{\lambda\mu + \lambda\nu + \mu\nu} \left\{ 1 - e^{-(\lambda + \mu + \nu)t/2} \left[\cos(t\sqrt{ \epsilon }) + \frac{1}{2\sqrt{ \epsilon }} \left(\frac{\mu^2 + \nu^2}{\mu + \nu} - \lambda \right) \sin(t\sqrt{ \epsilon }) \right] \right\}$ | $\frac{\lambda(\mu + \nu)}{\lambda\mu + \lambda\nu + \mu\nu} + \frac{\mu\nu}{\lambda\mu + \lambda\nu + \mu\nu} e^{-(\lambda + \mu + \nu)t/2} \left[\cos(t\sqrt{ \epsilon }) + \frac{\lambda + \mu + \nu}{2\sqrt{ \epsilon }} \sin(t\sqrt{ \epsilon }) \right]$ |
| | | $\epsilon = 0$ | $\frac{\lambda(\mu + \nu)}{\lambda\mu + \lambda\nu + \mu\nu} \left\{ 1 - e^{-(\lambda + \mu + \nu)t/2} \left[1 + \frac{t}{2} \left(\frac{\mu^2 + \nu^2}{\mu + \nu} - \lambda \right) \right] \right\}$ | $\frac{\lambda(\mu + \nu)}{\lambda\mu + \lambda\nu + \mu\nu} + \frac{\mu\nu}{\lambda\mu + \lambda\nu + \mu\nu} e^{-(\lambda + \mu + \nu)t/2} \left[1 + \frac{\lambda + \mu + \nu}{2} t \right]$ |
| | | $\epsilon > 0$ | $\frac{\lambda(\mu + \nu)}{\lambda\mu + \lambda\nu + \mu\nu} \left\{ 1 - e^{-(\lambda + \mu + \nu)t/2} \left[\cosh(t\sqrt{ \epsilon }) + \frac{1}{2\sqrt{ \epsilon }} \left(\frac{\mu^2 + \nu^2}{\mu + \nu} - \lambda \right) \sinh(t\sqrt{ \epsilon }) \right] \right\}$ | $\frac{\lambda(\mu + \nu)}{\lambda\mu + \lambda\nu + \mu\nu} + \frac{\mu\nu}{\lambda\mu + \lambda\nu + \mu\nu} e^{-(\lambda + \mu + \nu)t/2} \left[\cosh(t\sqrt{ \epsilon }) + \frac{\lambda + \mu + \nu}{2\sqrt{ \epsilon }} \sinh(t\sqrt{ \epsilon }) \right]$ |
| 4 | Repairable (faults detected upon inspection) $q = \lg(3 - \lg\theta\lambda)$ $\lambda_{eff} = 2\sqrt{1 - \Gamma(1/q)/q} \theta/n^2 + \lambda \frac{n - \theta}{n} \left(\frac{n - \theta}{n} + \frac{2\Gamma}{n} \right)$ | $t \geq m\eta_+$ | $1 - e^{-(t - m\eta)\lambda_{eff}} \left[1 - e^{-\left(\frac{t - m\eta}{\theta}\right)q} \right]$ | $0 \leq t < \eta_-$ 1 |
| | | $t \geq m\eta_+, m > 0$ | $1 - e^{-(t - m\eta)\lambda_{eff}} \left[1 - e^{-\left(\frac{t - m\eta}{\theta}\right)q} \right]$ | |
| | | $t = m\eta_-, m > 0$ | $1 - e^{-n\lambda_{eff}} \left[1 - e^{-(n/\theta)q} \right]$ | $t = m\eta_-, m > 1$ $1 - e^{-n\lambda_{eff}} \left[1 - e^{-(n/\theta)q} \right]$ |

ic test intervals and allowable test bypass times to meet unavailability goals for systems containing identical components arranged in one of the following logic configurations:

1-out-of-2; 1-out-of-2, twice; 2-out-of-3; or 2-out-of-4 logic. His assumptions are the same as Jacobs [Ja68] and Lofgren [Lo81].

To derive his nomographs, Hirsch apportions the unavailability goal equally between testing downtime and undetected standby failures. This procedure follows Section 4.11 of ANSI/IEEE Std 279-1971, which requires that the unavailability of the system due to test bypass must be commensurate with the unavailability of the system for the entire interval if no bypass were applied. He then uses the equations for unavailability due to undetected standby failures presented in what is now ANSI/IEEE Std 352-1975, Table 2, and the equations for unavailability during testing which he derives in his paper to develop nomographs for selection of the test interval and downtime which will meet a given unavailability goal for a given component standby failure rate. Figure 2.6 illustrates the use of his nomographs for a system configured with 2-out-of-3 logic.

It should be pointed out that unavailability will not be reduced by decreasing the test interval below the calculated value associated with the optimal unavailability without also decreasing the test bypass time. A shorter test interval means that the system will be bypassed more often, with a

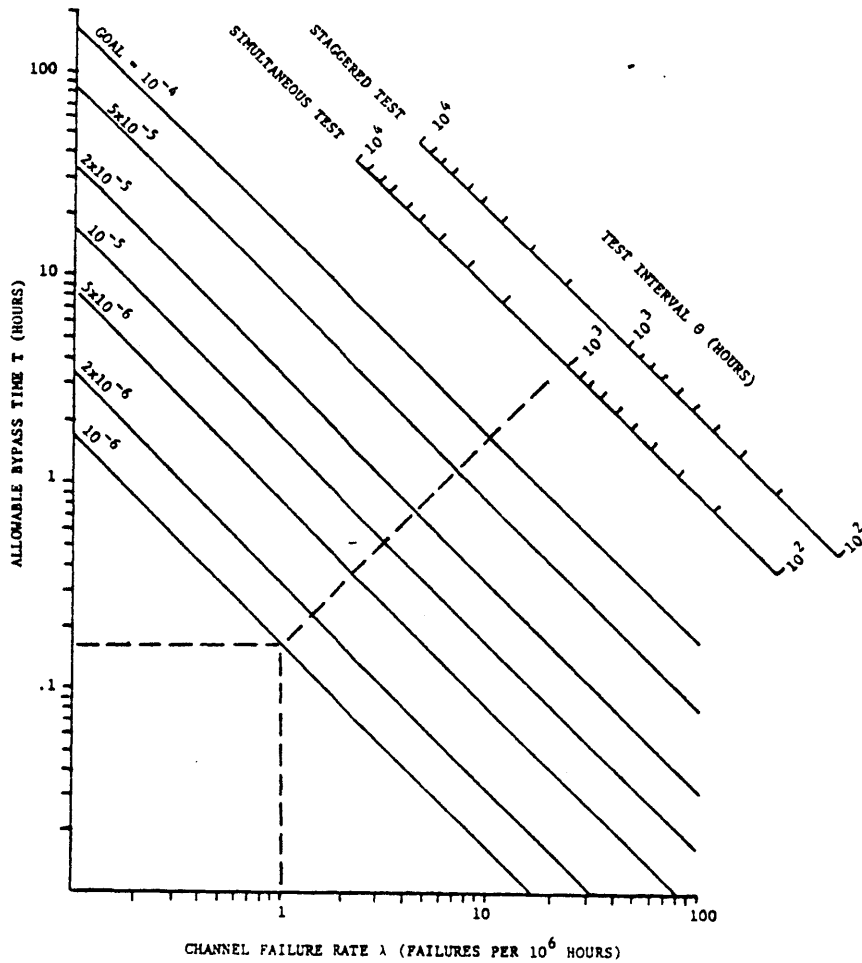


Figure 2.6. Nomograph for Calculating a Test Interval and Duration to Meet an Unavailability Goal. Example for 2-out-of-3:Good Logic. [Hi71]

commensurate increase in unavailability to override the periodic test. Also, the nomographs do not apply to groups of components for which the unavailability goal can be met before the testing contribution becomes equal to the failure contribution. For this situation, Hirsch's equations would have to be used directly.

Chay and Mazumdar [Ch75] consider periodic testing strategies to meet unavailability goals when the policy includes provisions to alter test intervals when one component of a redundant system is failed. Their assumptions for unavailability during testing are the same as in NUREG/CR-2158. However, they define separate test intervals for use when 1) no components are down and 2) one or more components are down. Also, both downtime for component testing and repair are explicitly modeled. Using these assumptions they derive sets of linear equations to calculate the average cycle unavailability of 1-out-of-2:Good and 2-out-of-3:Good logic configurations containing identical components. They then apply the equations to a typical reactor trip system. Their equations are too complex to be presented outside the context of their paper. However, it is worth noting that for their examples a wide range of testing options will satisfy the safety goal.

Effects of Human Error on Simple Systems

McWilliams and Martz [McW80] have investigated the

effects of two types of human error on the optimum test interval of simple one component systems. They define the errors as:

- Type A Human Error - An initially operational standby safety system is inadvertently left in an undetected failed state at the conclusion of the test.
- Type B Human Error - A system failure is not detected by an inspection which should have revealed it.

The authors develop a Markov model for the steady-state availability of a simple system subject to these errors using the same assumptions as NUREG/CR-2158, with the exception that a time dependent failure rate is allowed. The authors use a Weibull hazard function for their example.

With their model the authors calculate the sensitivity of the optimum test interval, τ^* , and the resultant availability, A^* , at that interval to the probability of Type A and B human error, p_A and p_B , respectively. Their results are shown in Figures 2.7. and 2.8 for a component having a Mean Time To Failure of 100 time units, a shape factor of 2.0, and test downtime of 1 hour. The results have two limitations. First, failures or human errors that generate a requirement for repairs that extend the downtime, are not considered. (All repairs are assumed to be accomplished instantaneously.) Second, although sensitivity of the optimal test interval to changes in human error probabilities are presented, the sensitivity of availability to deviations from the optimum was not shown, as was

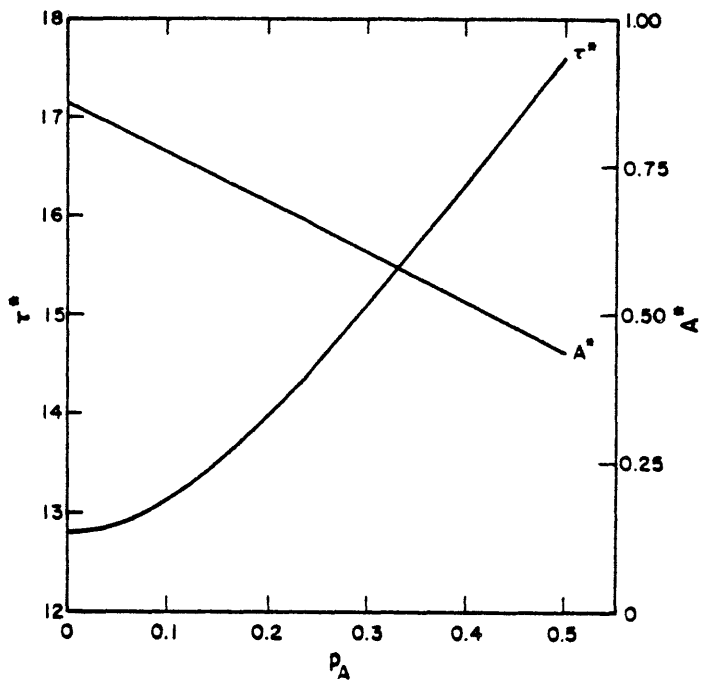


Figure 2.7. Optimal (Maximum Availability) Test Interval and Associated Availability as a Function of p_A . [McW80]

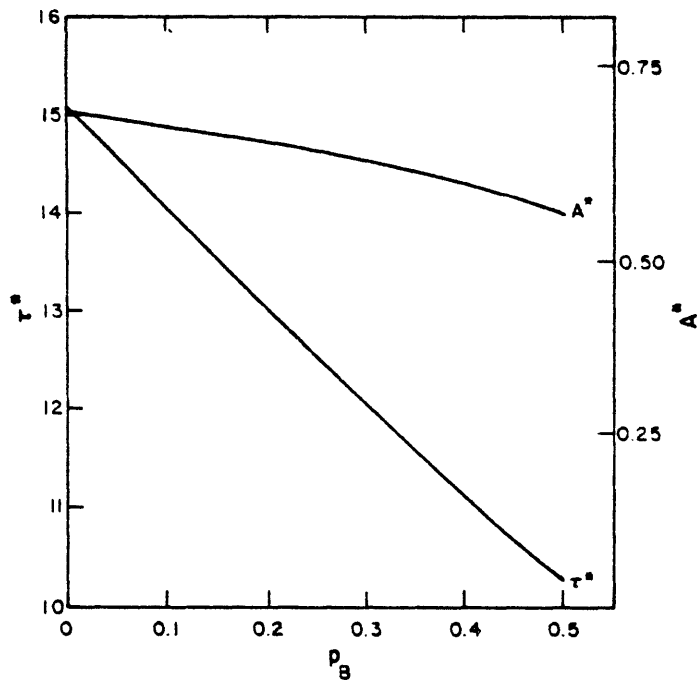


Figure 2.8. Optimal (Maximum Availability) Test Interval and Associated Availability as a Function of p_B . [McW80]

done in NUREG/CR-2158. It is also interesting to note that the optimum test interval changes by less than one time unit as either human error probability increases from zero to 10%. From a practical point of view, if an analyst were to determine that the error probability is 10%, he should find a better way to accomplish the test rather than to recommend that the test interval be lowered slightly to maintain an optimum test interval.

In a subsequent work, [McW81] McWilliams expands the work described above to account for two additional types of human error:

Type C - Improper repair of a failed component, and

Type D - Failure to locate the reason for an annunciator-activated inspection.

McWilliams accomplishes a sensitivity study with all four types of human error and concludes that Type A human errors, which leave the component in an undetected failed state, produce significantly increased unavailability as it increases. This is not unexpected, since the probability that the error is committed becomes the minimum probability that the component will fail upon demand during the entire standby period. The other types of errors have little effect on unavailability. McWilliams also comments on the benefits of annunciating failures. Unfortunately many important components, by the fact they they are idle, are incapable of annunciating their failures.

Apostolakis and Bansal [Ap77] consider human error during testing which leaves a component in an undetected failed state (defined as a Type A error by McWilliams and Martz [McW80]). Specifically, they investigate the importance of dependencies among human errors committed in sequential inspections of identical components. Their unavailability formula account for:

λ \equiv Failure modes with a constant standby failure rate

q_D \equiv Demand failure modes

τ \equiv Standby interval (\approx the test interval when $\tau \gg n\tau_r$, where $n\tau_r$ is the total time required to test n components sequentially.)

τ_r \equiv Component downtime for test, maintenance and repair.

γ_0 \equiv Probability that operator error leaves a component in an undetected failed state for the first time during a series of sequential inspections.

γ_j \equiv Conditional probability of the human error being repeated for the $(j+1)$ time given that it has occurred for j consecutive times in the current inspection period

The authors derive equations for q_r , average unavailability due to hardware failures only, and q_h , average unavailability when at least one component has been failed by human error. Equations are presented for a number of common logic configurations. Those for a 1-out-of-2:Good system are given here as an illustration:

$$q_r = \frac{(\lambda\tau)^2}{3} + Q_D(\lambda\tau) + Q_D^2 + \frac{\tau_r}{\tau}[\lambda\tau + 2Q_D] \quad (2.14)$$

$$q_h = \gamma_o \gamma_1 + 2\gamma_o [(\lambda\tau)/2 + Q_D] + 2\gamma_o(\tau_r/\tau) \quad (2.15)$$

These equations may be interpreted as follows: For a 1-out-of-2:Good system to be unavailable, both components must be failed simultaneously. Unavailability due to hardware failures can occur because of 1) two standby failures, 2) one standby and one demand failure, 3) two demand failures, or 4) one component is being tested when the other has failed. Human error unavailability can occur because of 1) two sequential human errors, 2) human error combined with either a standby or demand failure, or 3) human error has failed one component and the second is being tested.

The Apostalokis and Bansal paper also discusses the concept of conditional human error probabilities. Complete independence of sequential errors implies that:

$$\gamma_o = \gamma_1 = \gamma_2 = \dots \quad (2.16)$$

Complete dependence implies:

$$1 = \gamma_1 = \gamma_2 = \dots \quad (2.17)$$

Under the assumption of complete dependence, an operator who errs during first of a number of sequential inspections (γ_o)

will repeat the error during each subsequent inspection with a probability of one.

The degree of dependence will actually be somewhere between the two extremes, in which case the human error probabilities will have the bounds $\gamma_0 \leq \gamma_j \leq 1$, for all j . For example, one might judge that the probability of making an error during the test of a component might be $\gamma_0 = 0.001$, but, given there are circumstances under which an operator would make that error, the probability of him making it again while testing a second component might be $\gamma_1 = 0.25$. The combined probability of leaving both components of a 1-out-2:Good system failed due to human error would then be $2.5E-4$.

The effect of human error is to reduce the sensitivity of system unavailability to test interval. The failures remain undetected until the next periodic test, when they are assumed to be detected. That second test can fail to detect the failure or produce its own. Consequently, the human error probabilities remain the same from test to test and are independent of test interval. They are not independent of testing strategy, however. For example, staggered testing has the potential of greatly reducing the probability of dependent human errors. Provided the test procedure is correct, the chance of making the same mistake over again when the tests are two weeks apart should be smaller than when the tests immediately follow each other. Apostalokis and Bansal in essence make a strong argu-

ment for accomplishing staggered testing rather than sequential testing.

In a subsequent work, [EP1443] Apostolakis, Chu and Whitley expand on the above work and present much of the background and reasoning behind their models. The work contains an excellent literature review, a discussion of failure mechanisms, and common cause failure modeling, both from a hardware and maintenance point of view. They develop unavailability equations for a number of common component configurations and compare their results with FRANTIC calculations, obtaining good agreement.

Components With Many Failure Mechanisms

Research which accounts for failure mechanisms that affect the time dependent unavailability of components in a variety of ways will now be reviewed. These models are improving knowledge of the relative importance of various types of failures to the overall unavailability of standby systems. The FRANTIC models of component failures mechanisms fall into this category. The FRANTIC code is introduced in the next section and will be discussed in detail in Chapter 3.

Demand related failures, which are independent of the test interval of a standby component, can dilute the effects of periodic testing. In two recent papers Mankamo [Ma81] and Mankamo and Pulkkinen [Ma82] have addressed the division of observed failure between standby and demand related mechanisms. In this work they were looking at U.S. Licensee

Event Reports (LER) to gain information about dependencies in the failure of diesel generators. The classification between standby and test observed demand failures was accomplished by fitting the observed failure frequency of generators verses test interval to the relation:

$$q(T) = q_d + \lambda T \quad (2.18)$$

Where: (Notation changed)

$q(T)$ - Observed failures per test demand

q_d - Transition failure rate

λ - Standby failure rate

T - Periodic test interval

Their results are given in Figure 2.9. It can be seen from this graph that for test intervals greater than about one week random standby failures tend to dominate observed demand failures, the two values being $\lambda = 8.7E-5/\text{hr}$ and $q_d = 6.6E-3$. Using this graph Mankamo suggests that test intervals from 1 to 4 weeks appear to be reasonable. It should be noted that the demand failure rate can model conditions of a true demand which cause failure as well as those which occur at tests. Consequently, the shorter test intervals may not produce the unavailability suggested by Figure 2.9.

Finally, as part of the dependency investigation, Mankamo obtained time dependent hazard rates for diesel generator failures. He found that a fast aging contribution attributable

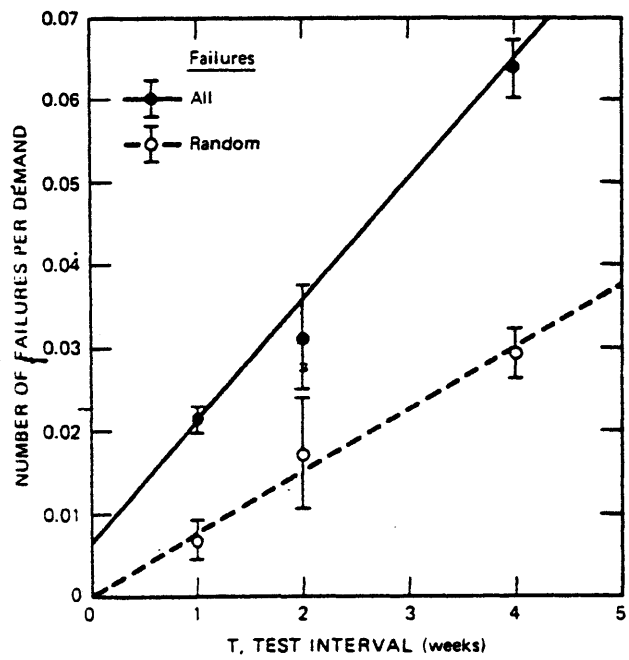


Figure 2.9. Test Interval of Diesel Generator Failures. ("All" refers to dependent and random failures.) Confidence bounds represent 90% intervals (statistical variation only). [Ma82]

to design errors could be modeled adequately by a Weibull hazard rate.

Signoret [Si79] has developed a model for time dependent and average unavailability of standby safety systems which accounts for:

- Periodic test duration of π hours in which the system may be either unavailable or available.
- Standby failures which occur with a rate of λ per hour.
- Standby period of ϕ hours duration.
- Transition failures which occur with a rate of γ per demand.
- Active mode failures which can occur during testing with a rate of λ' per hour.
- Downtime for repair which has a completion rate of μ per hour.
- Test interval of τ hours between the beginning of successive periodic tests.

Signoret derives both instantaneous unavailability and time averaged steady-state unavailability expressions for a one component system. He defines two cases:

- 1) Component is not available for its safety function during the test.
- 2) Component is available for its safety function during the test.

His expressions for the average unavailability are:

$$Q1 = \frac{\pi}{\tau} + \frac{\lambda\phi}{\mu\tau} + \frac{\gamma_t}{\mu\tau} + \frac{\lambda\phi^2}{2\tau} \quad (2.19)$$

$$Q2 = \frac{\pi}{\tau} + \frac{\lambda\phi}{\mu\tau} + \frac{\gamma_t}{\mu\tau} + \frac{\lambda\phi^2}{2\tau} + (1-\gamma)\frac{\lambda'\pi^2}{2\tau} + \frac{\lambda\pi\phi}{\tau} \quad (2.20)$$

Where:

$\gamma_t = \gamma + (1-\gamma)\lambda'\pi \equiv$ total probability that the component fails because of the test.

The above equations can be interpreted in terms of probabilities of specific failures lasting for specific periods of time. The τ in the denominator of each term appears because the equation is averaged over a test interval. The terms may be interpreted as follows:

- π - Time integrated unavailability of a component which not able to perform its function during the test period (Case 1).
- $\gamma\pi$ - Time integrated unavailability due to demand failures occurring at the test (Case 2). It is assumed that the demand failure will last for the duration of the test period.
- $(\lambda\phi)/\mu$ - Time integrated unavailability due to the repair of standby failures detected during the test. The quantity $(1/\mu)$ is the average duration of the test.
- γ_t/μ - Time integrated unavailability due to the repair of failures caused by the test.

- $(\lambda\phi^2)/2$ - Average time integrated unavailability due to undetected standby failures times the standby period.
- $(1-\gamma)(\lambda'\pi^2)/2$ - Average time integrated unavailability due to active mode failures during the test (Case 2).
- $\lambda\phi\pi$ - Time integrated unavailability of standby failures which existed at the beginning of a test and continue to exist throughout the test period.

An additional term, $(\gamma\phi)/\tau$, should be added to both equations to account for the fact that demand failures can occur when the system responds to an accident during the standby period as well as when it is being started for testing. Signoret did not account for this possibility in his derivation. Also, since the above equations are averaged over the entire test cycle, the unavailability obtained by using them in conjunction with fault trees may be unrealistic. To be flexible, equations for the unavailability of periodically tested components should be broken into the specific time frame when they are under test, repair, or standby.

Vaurio [Va79a, Va80, Va82] and Vaurio and Sciaudone [Va79b] have developed models of component failure mechanisms which contain the most comprehensive set of failure mechanisms which this author has encountered. The models include (taken from [Va82]):

λ_s - Failure rate during standby periods

η - Test interval, time between inspections (T_2 in FRANTIC

II-MIT)

- v - Test duration (τ in FRANTIC II-MIT)
- τ - Repair duration for a failed component (T_R in FRANTIC II-MIT)
- q_o - Fraction of v that component is down during a test, or test override unavailability
- p - Probability of failure due to test, failure repaired after the test (P_f in FRANTIC II-MIT)
- γ - Probability of failure due to a test, failure not repaired (detected) before the next test (part of q_d in FRANTIC II-MIT)
- ρ - Probability of failure due to a true demand (e.g., an event exceeding the design basis criteria of a component) (part of q_d in FRANTIC II-MIT)⁴
- ω - Probability that a failed component is not detected by a test or not repaired (q_u , C_f , or q_d in FRANTIC II-MIT depending on failure)
- p_B - Probability that a periodic test or inspection fails to detect a failed component (q_u or C_f in FRANTIC II-MIT)
- P_I - Fraction of random failures detectable by a continuously monitoring annunciator system (modeled as separate components in FRANTIC II-MIT)

⁴ The demand failure rate in FRANTIC II-MIT generates a repair unavailability implying failures during periodic tests, so it is not an exact model of event generated failures. See Section 3.3.

β - Probability that a monitored failure will not be detected until next test (converts monitored λ into periodically tested λ)

In [Va79b] Vaurio and Sciaudone apply components with these failure mechanisms to redundant m-out-of-n:Good systems up to m=n=4 to determine average system unavailability verses periodic testing policies which include sequential, staggered, and random testing. They have compared their results with the original FRANTIC code and found agreement for most systems. However, they found large discrepancies between FRANTIC and ICARUS, a computer code they developed to calculate the unavailability equations they developed for the system in question, for systems with high redundancy which are tested sequentially. It is believed that the discrepancy occurred because FRANTIC rounds off t_1 to the nearest hour. This variable establishes the staggering times of the various periodic tests, and the round off causes inadvertant test overlap. This problem has been corrected in FRANTIC II-MIT and is discussed in more detail in Chapter 5.

In [Va82] Vaurio derives an equation very similar to those used to calculate the optimum test interval, η^* , of single components in subroutine OPTTEST, which is presented in Chapter 4. His equation is:

$$\eta^* \approx \left[\frac{2 [q_{0v} + (p+\gamma) \tau]}{(1-P_I + \beta P_I) \lambda_S} \right]^{\frac{1}{2}} \quad (2.21)$$

In this equation $(1-P_I + \beta P_I)$ is simply the fraction of standby failures which are detected by periodic tests. In FRANTIC II-MIT this same effect is obtained by defining a periodically tested component so that standby failures can be detected only at the test. This equation also assumes that test caused failures, modeled here by p , are accounted for as part of q_0 during the test. It neglects the possibility that random failure which should be detected by a periodic test is left undetected through the next standby interval. This is the Type A human error modeled by McWilliams and Martz [McW80] and accounted for by C_f in FRANTIC II-MIT.

Vaurio's work covers a wide range of topics important to time dependent unavailability analysis, including the effects of common-mode and undetected failures in redundant systems. Although much of his work parallels the models contained in FRANTIC II-MIT, it is still limited by system specific unavailability equations which leave little flexibility to investigate the effects of changes in design, especially where diverse safety functions are possible. For this type of work a computer code which can be easily used with a cut set generator is required.

2.3.3 APPLICATION TO FAULT TREE ANALYSIS

It is very difficult to evaluate the effect of a testing

policy for a specific component or group of components within the context of a larger system having many different types of components. Often, because of the diversity of their functions, all the parts of a system can not be tested at once. The degree by which the conditions of the accident can be simulated will vary according to component location, function, and expected environmental conditions during an accident. Each component can have both standby and demand failure mechanisms contributing to a particular failure mode. The relative importance of individual components will vary depending on their function within the system and the system's safety function for a given accident sequence. Because of these difficulties, the effects of periodic testing and maintenance are frequently analyzed manually using models such as those discussed above for the particular application of interest. The results are then time averaged and applied in the fault tree as constant per demand failure modes.

A typical set of time averaged unavailabilities might be:

$$q_{av} = q_{hardware} + q_{test} + q_{maint} + q_{hep} \quad (2.22)$$

Where:

$$q_{hardware} = (\lambda T)/2 + q_d \quad (2.23)$$

T \equiv Periodic test interval

q_d \equiv Demand failure rate

$$q_{\text{test}} = \tau/T \quad (2.24)$$

τ \equiv Test downtime

$$q_{\text{maint}} = f(t_D) \quad (2.25)$$

f \equiv Frequency of unscheduled maintenance and repair.

t_D \equiv Average time to complete the unscheduled maintenance.

q_{hep} = Unavailability due to human error.

The unavailability q_{hep} is usually estimated from human error models, such as those presented by Swain and Guttman [Sw80].

The time averaging method has the disadvantage of masking combinations of high instantaneous unavailabilities which can combine to produce a large system unavailability for some period of time. For example, a component whose instantaneous unavailability is 0.001 will have very little effect on system unavailability when it is parallel with a component which has just been tested and is known to be working. However, if the second component is completely unavailable due to testing, the instantaneous system unavailability will be 0.001. Because of the complexity of most practical systems, a computer code is required for a time dependent analysis of their unavailability.

2.3.4 THE FRANTIC II COMPUTER CODE

The Nuclear Regulatory Commission has recently released a

computer code which can alleviate the problems involved with analyzing the unavailability of complex systems. The most recently released version is FRANTIC II. [Ve77, Ve81] A major feature of the code is its ability to account for the effects of imperfect testing through the use of a variety of component input parameters. It can also model time dependent failure rates, both as a function of calendar time and and test frequency. The code calculates the instantaneous unavailability of every component in the system before and after each time point at which any component might have a discontinuous jump in its unavailability. (These times correspond to passage from standby to active testing to repair of failures found during testing in periodically tested components.) It then calculates the system unavailability at each time point with a user supplied unavailability equation and time averages the instantaneuous system unavailabilities over the calculation period. It outputs the average system unavailability over the calculation period and (optionally) the instantaneous unavailabilities at each time point. Through this process FRANTIC II avoids the need for deriving cumbersome formulas for average system unavailability. Any system whose failure can be described by a coherent fault tree can be quantitatively analyzed using FRANTIC II. When interfaced with a cutset generation and evaluation routine the FRANTIC II code becomes a versatile tool for investigating the unavailability of a complex system as a function of its periodic testing policy. An

engineering interpretation of this code and the modifications accomplished on it as part of this work is presented in Chapter 3.

CHAPTER 3

ENGINEERING INTERPRETATION OF FRANTIC II-MIT

This chapter presents an engineering interpretation of the FRANTIC II-MIT code. First, the overall structure of the code is introduced. Simplified equations show how the various possible component and test failure mechanisms contribute to a component's unavailability. This is followed by a brief description of how the code uses input to accomplish its calculations. Next, the input parameters to the code are interpreted in terms of the physical failure mechanisms they can represent. Limitations imposed by the way the code calculates with a particular parameter are discussed and suggestions are made for ways to represent common modeling problems that the systems engineer might encounter. The estimation of input parameters that represent time dependent failure rates is presented in Section 3.6.

FRANTIC II-MIT follows the basic structure of FRANTIC II, but with the following additions and corrections:

- The code has been interfaced with a code that generates and evaluates cut sets from fault tree logic. This package functions as the user supplied SYSCOM Subroutine.
- The Weibull hazard rate has been generalized by the addition of an offset time (defined in Section 3.6.2).

- The demand failure rate can now be changed by periodic testing to reflect potential wear-out or burn-in mechanisms.
- A factor has been added to account for human error which fails to detect standby failures.
- Undetectable failures are allowed to accumulate and are unaffected by tests under the code's New-New component renewal type (See Section 3.2.4 for a description of the New-New option.).
- The first test interval, T_1 , is no longer rounded off to the nearest hour. This prevents potentially large errors in sequentially tested redundant components.
- A subroutine has been added to calculate the optimum test interval of a single component.
- Provision to write directly to user formatted files while suppressing the standard output has been added.

Although this list may seem long, none of them change the overall structure of the code, which is presented in the next section. They either make the code's modeling capability slightly more flexible or make it more convenient to use. Of all these changes, the offset time probably adds the most flexibility to the code, since it allows the user to project the effects of component wearout years into the future without the requirement of running the code through all the intermediate years. This overcomes the time point limitations imposed by storage dimensioning requirements.

3.1 OVERALL STRUCTURE

3.1.1 UNAVAILABILITY EQUATIONS

FRANTIC II-MIT uses two sets of equations to calculate component unavailability. The equations are essentially the same as those used in FRANTIC II with the additional options mentioned in the introduction to this chapter. For periodically tested components,¹

During Test Period n,

$$\begin{aligned} q_1 = & q_d + (1 - q_d)Q_n + (1 - q_d)(1 - Q_n)q_o \\ & + (1 - q_d)(1 - Q_n)(1 - q_o)P_f \\ & + (1 - q_d)(1 - Q_n)(1 - q_o)(1 - P_f)q_u \end{aligned} \quad (3.1)$$

During the repair period following Test n,

$$\begin{aligned} q_2 = & q_d + (1 - q_d)Q_n + (1 - q_d)(1 - Q_n)P_f \\ & + (1 - q_d)(1 - Q_n)(1 - P_f)q_u \\ & + (1 - q_d)(1 - Q_n)(1 - P_f)(1 - q_u)q_\lambda(T_R/2) \end{aligned} \quad (3.2)$$

During standby following Test n,

¹ These equations are approximate. The precise equations depend on renewal type. See the FRANTIC II Manual and the listing of FRANTIC II-MIT, which is well documented with comments.

$$\begin{aligned}
q(t) = & q_d + (1 - q_d)q_\lambda + (1 - q_d)(1 - q_\lambda)q_\mu \\
& + (1 - q_d)(1 - q_\lambda)(1 - q_\mu)C_f q_\lambda(T_2)
\end{aligned}
\tag{3.3}$$

For all other types of components,

$$q_{av} = q_d + (1 - q_d)q_m + (1 - q_d)(1 - q_m)q_\mu
\tag{3.4}$$

Where:

- q_d - Unavailability due to demand, human error, and transition failures
- q_λ - Unavailability due to detectable standby failures
- Q_n - Probability of a detectable failure at Test n
- q_μ - Unavailability due to undetectable/unreparable standby failures
- P_f - Probability of the test causing a failure that requires repair
- q_o - Unavailability of a good component due to test and maintenance
- C_f - Probability of not detecting a detectable standby failure (represented by $q_\lambda(T_2)$) at the last test
- q_m - Average unavailability of monitored components due to standby failures

Factors, such as $(1 - q_o)$ are included to account for the fact that the failure modes are mutually exclusive. They may be interpreted as, "If the component has not already

become unavailable because it has been taken off line for testing, then...."

Three equations are used for periodically tested components because of the known cycle of standby, test and repair. The term q_λ represents transitions to a failed state which have occurred, but because of the idleness of the component have not yet been detected. When a test is accomplished those failures are revealed, with the probability Q_n of the component being in a failed state at the test.

Although not shown explicitly in the above equations, q_λ may be reset to zero at two times, the end of test and the end of repair. The split between the two depends on the probability that the component required repair at the last test. See the FRANTIC II Manual for a detailed description of these equations.

Monitored components are the most common type of component which are not periodically tested. Their transitions into failed states are detected when they occur, or at some random time after they occur. The term q_m accounts for both the rate at which the failures occur and the average time between failure and return to a working state.

The terms q_d and q_u are common to all types of components. Their physical interpretation is presented below. However, they can also be used to conveniently represent any unavailability that is not reset by a periodic test. A more

complete explanation of the input parameters which will calculate the above unavailabilities is given in Sections 3.3 through 3.6.

3.1.2 CALCULATION PROCEDURE

FRANTIC II-MIT follows the FRANTIC II calculation procedure exactly. It is a series of subroutines driven by a main program which is controlled by keywords and formatted input. (Because additional failure parameters have been added, the format for the COMPONENT data group is entirely different from that of FRANTIC II. See Appendix I.) The user may input data for any number of calculations that he desires. After the code has completed the calculations generated by one set of keyword input, it will automatically shift to the next. Figure 3.1 briefly describes the computational flow of the code.

A major feature of FRANTIC II-MIT is its ability to account for discontinuities in the time dependent system unavailability. SUBROUTINE TIMES determines all the time points at which periodic tests or repair are started or completed. SUBROUTINE QCOMP then calculates the unavailability of each component just before and just after the time point and calls the user supplied SUBROUTINE SYSCOM to calculate the system unavailability. This subroutine contains the equation resulting from the quantitative evaluation of the system fault tree, which is not accomplished by FRANTIC

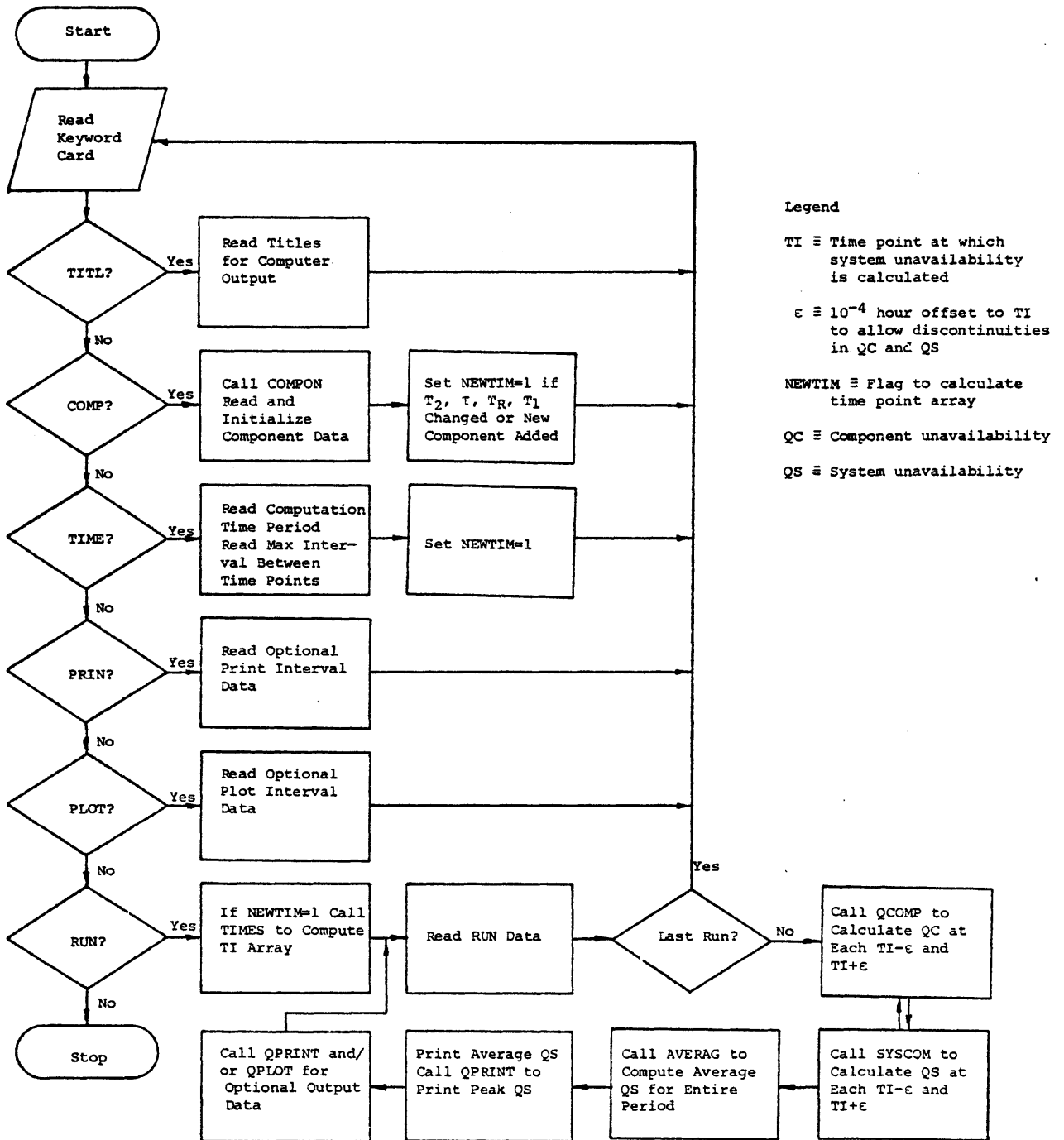


Figure 3.1. Computational Flow of the FRANTIC Computer Programs.

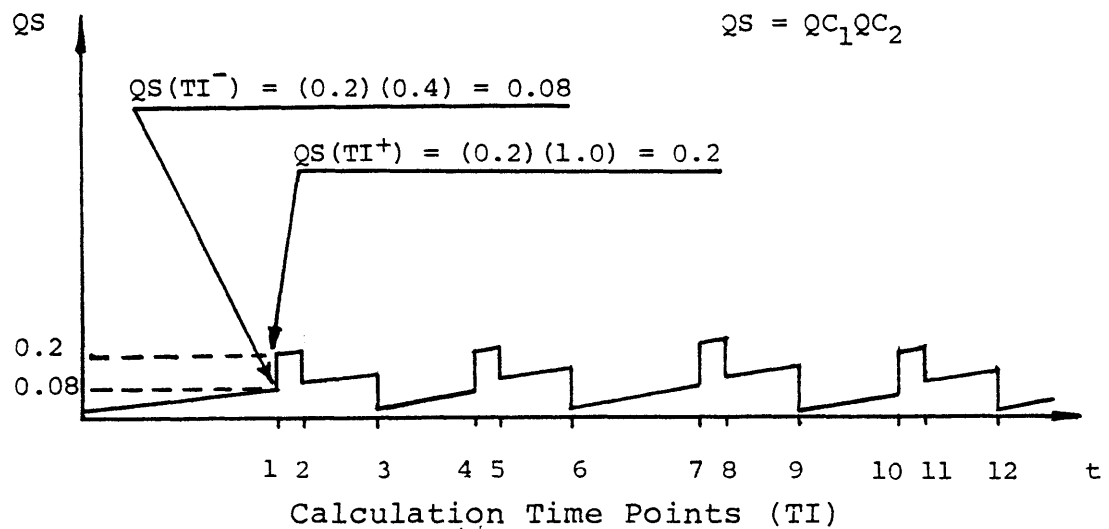
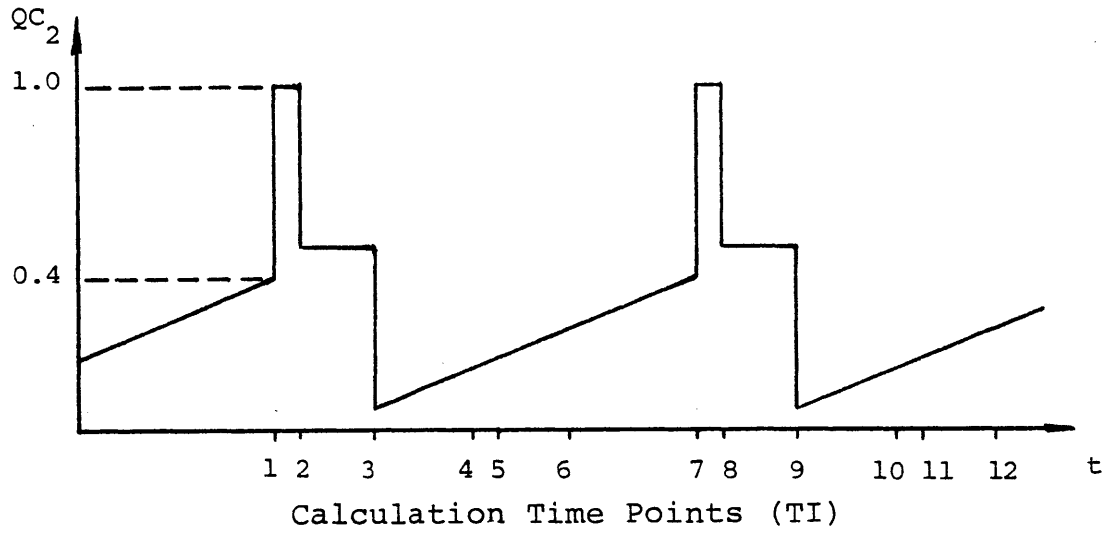
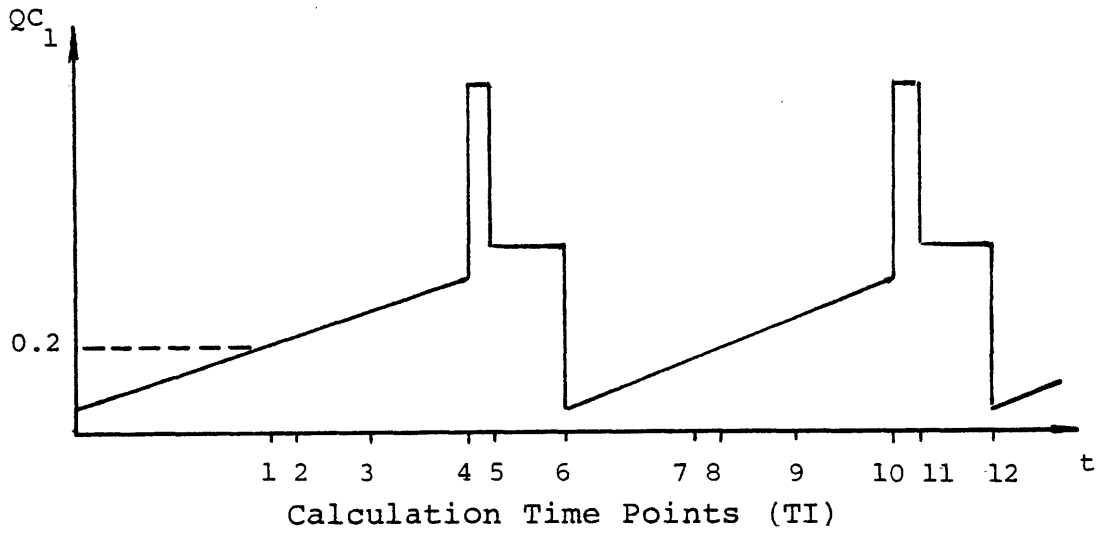


Figure 3.2. Example of FRANTIC's Use of Time Points to Calculate the Instantaneous Unavailability of a Two Component Parallel System.

II-MIT. Figure 3.2 illustrates the way FRANTIC II computes the unavailability of a simple 2 component parallel system.

When system unavailabilities before and after each time point have been calculated, SUBROUTINE AVERAGE time integrates the unavailability by assuming unavailability varies linearly between the time points. The subroutine then divides by the time interval of the calculation to obtain the average system unavailability.

FRANTIC II-MIT contains the print and plot subroutines of FRANTIC II. With these one obtains formatted output of the average system unavailability over the calculated time period, unavailabilities, instantaneous system unavailabilities, including an ordering of the largest, average system unavailability during intermediate time intervals, and plots of the time dependent system unavailability. (The plot routine must be updated to interface with local plotting software.) SUBROUTINE FILES with an option to bypass the standard formatted output and write selected output in a more compact form to user designated files is an additional modification in FRANTIC II-MIT.

The portion of FRANTIC II's output which breaks the average system unavailability into contributions arising from testing, repairs and failures should be treated with caution. The FRANTIC II Manual calls them contributions "due to" testing repair and failures. However, this use of words is very misleading. The rules for apportioning

instantaneous system unavailability to one of the three categories is as follows: [VE81]

- 1) If at least one component is under test then the instantaneous system unavailability is counted toward the test contribution.
- 2) If no components are under test and at least one component is down for repair, then the instantaneous system unavailability is counted towards the repair contribution.
- 3) If no components are under test or repair, the instantaneous system unavailability is counted towards the failure contribution (i.e., between test contribution)

What is actually listed under the three categories is the system unavailability due to various causes during specifically defined time periods. And it is not an average system unavailability over just those specific time periods, but more precisely the time integrated unavailability over the specific periods divided by the total calculation time. The unavailability given under each category therefore has no physical meaning. The only numbers that can be interpreted are the percentage figures. For example, the test percentage can be interpreted as the percentage of the average total system unavailability over the total calculation time which accumulates while at least one component is under repair. To avoid this misinterpretation, the output of FRANTIC II-MIT has been modified as shown in Figure 3.3.

***** SYSTEM UNAVAILABILITY DATA *****

 RUN OPTIONS:

| SYSTEM DESCRIPTION | EQUATION NUMBER | UNAVAIL. OPTION | PLOT OPTIONS X Y | CUTOFF OPTION |
|--|-----------------|-----------------|------------------|---------------|
| STAGGERED AT TEST INTERVAL, DEL TIME = 0.5 DAY | 1 | TOTL | NONE NONE | 0 |

 SYSTEM MEAN UNAVAILABILITIES BETWEEN ZERO AND 363.00 DAYS:

| TOTAL MEAN UNAVAIL | UNAVAIL DURING STANDBY | % OF TOTAL | UNAVAIL DURING TESTING | % OF TOTAL | UNAVAIL DURING REPAIRS | % OF TOTAL |
|--------------------|------------------------|------------|------------------------|------------|------------------------|------------|
| 3.165E-09 | | 0.61 | | 97.29 | | 2.10 |

 PEAK SYSTEM UNAVAILABILITIES:

| POINT NUMBER | TIME (DAYS) | TIME (HOURS) | SYSTEM UNAVAIL |
|--------------|-------------|--------------|----------------|
| 77 | 3.0375E+01 | 7.2900E+02 | 6.0603E-07 |
| 148 | 6.0375E+01 | 1.4490E+03 | 6.0592E-07 |
| 219 | 9.0375E+01 | 2.1690E+03 | 6.0592E-07 |
| 290 | 1.2037E+02 | 2.8890E+03 | 6.0592E-07 |
| 361 | 1.5037E+02 | 3.6090E+03 | 6.0592E-07 |
| 432 | 1.8037E+02 | 4.3290E+03 | 6.0592E-07 |
| 503 | 2.1037E+02 | 5.0490E+03 | 6.0592E-07 |
| 574 | 2.4037E+02 | 5.7690E+03 | 6.0592E-07 |
| 645 | 2.7037E+02 | 6.4890E+03 | 6.0592E-07 |
| 716 | 3.0037E+02 | 7.2090E+03 | 6.0592E-07 |
| 787 | 3.3037E+02 | 7.9290E+03 | 6.0592E-07 |
| 858 | 3.6037E+02 | 8.6490E+03 | 6.0592E-07 |

To if every total system unavailability over the total calculation time which occurred while at least one component was repaired.

Figure 3.3. FRANTIC II-MIT Output of System Unavailability Data Resulting From a Calculation Using the Run Option.

3.2 STANDBY FAILURE RATE

Both the detectable and undetectable standby failure rates are represented by a generalized Weibull hazard rate; given the component is not failed at time t , the probability that it will fail between t and $t+dt$ is,

$$P(\text{Fail}) = \lambda(t)dt = \beta\lambda_n(t+t_o-t_r)^{\beta-1}dt \quad (3.5)$$

Where:

$\lambda(t)$ - Conditional Failure Rate or Hazard Rate (sometimes designated by $z(t)$ or $h(t)$).

λ_n - Scale Factor.² The probability of failure is proportional to the scale factor, as it establishes the length (or scale) of time the component is expected to function before it fails. The subscript n specifies that n periodic tests have been accomplished prior to the current standby period.³

β - Shape Factor. The shape factor is used to specify how the failure rate varies with time (thereby determining its shape). It can be any value greater than zero which, in combination with t_o , produces the time dependence in $\lambda(t)$ which best matches that

² The scale factor after n tests is automatically calculated by the code based on a test caused change factor and component renewal type. See Sections 3.2.5 and 3.5.3.

³ The symbol for the scale factor has been changed to make it more distinguishable from the Conditional Standby Failure Rate.

obtained from failure data and/or engineering judgment.

t_r - Renewal Time.⁴ The renewal time represents the time at which the component is either reconditioned or replaced. When renewal occurs, the hazard rate time is reset to t_0 .

t_0 - Offset Time. This parameter has been added to the failure rate in FRANTIC II-MIT to give the analyst more flexibility in modeling time dependent failure rates. It establishes the time point on the hazard function curve at which the failure rate of the component immediately after renewal will be calculated. When it is negative, the hazard rate corresponds to the standard Weibull three parameter hazard rate, and the failure rate is zero until $(t+t_0)$ becomes positive. It will be discussed in detail in Section 3.6.

The remainder of this section discusses the physical interpretation of standby failure input parameters. Section 3.6 presents a more detailed development of the generalized Weibull hazard rate, including methods for estimating t_0 , β , and λ_0 when the time dependence of the conditional failure rate is known.

⁴ This parameter is automatically calculated by the code based on component renewal type. See Section 3.2.4.

3.2.1 SCALE FACTOR FOR DETECTABLE STANDBY FAILURES

The Scale Factor for Detectable Standby Failures, λ_n , models detectable failure mechanisms that occur during a component's standby period. The scale factor establishes the magnitude (in conjunction with β and t_0) of the probability that a detectable fault will occur per unit time at any given time. The scale parameter is used with both monitored and periodically tested components. For periodically inspected components the user inputs the scale factor for the initial standby interval, λ_0 . When tests cause wear-out, the code automatically calculates λ_n using λ_0 and f_λ . See Section 3.5.3 for a discussion of test caused changes in the scale factor.

Monitored components normally perform some kind of function while the safety system is on standby. Failures are either immediately announced in the control room or detected a short time later during normal operator rounds. The failure rate can therefore represent both internal hardware failure mechanisms and the effects of external shocks. Examples of monitored components are power supplies and sensors. The output voltage of a power supply can be constantly monitored, and shorts will be detected immediately. The outputs of many sensors are checked on a routine basis during the standby period. At this time malfunctions or suspicious output can be checked and repair affected if necessary.

Periodically tested components are normally idle during standby and must transfer to an active state when a demand

occurs. For these types of components, the scale factor models failure mechanisms which occur during the standby period, but which are not revealed until the component is required to operate. Since the component is idle, these failures occur primarily due to external random "shocks" resulting from the standby environment. Examples would be exposure to vibrations, process fluid flow, moisture, and human errors of commission during the standby period. A more specific example might be a leak which soaks the windings of an electric motor so that it will short out when called upon to start and run.

3.2.2 SCALE FACTOR FOR UNDETECTABLE STANDBY FAILURES

The Scale Factor for Undetectable Standby Failures, λ_{μ} , models failure mechanisms which occur during the standby period that will cause the component to fail to perform its safety function when called upon during a true demand, but which are not revealed by monitoring or periodic tests. It establishes the magnitude (in conjunction with β and t_0) of the probability that an undetectable fault will occur per unit time.

The general causes of this type of failure mechanism are the same as for λ_0 . However, periodic testing does not reveal these failures. Although the component apparently performs its safety function during a test, it fails during a true demand due to a failure mechanism not addressed by the test. Undetectable faults can occur when periodic tests require reconfiguration of the safety system from its operational alignment. For example, during an operational test, the High

Pressure Coolant Injection System's discharge is routed to the condensate storage tank. An obstruction beyond the test bypass line would not be detected by the test. A second example is the inability to simulate the environmental conditions of the accident during an operational test.

Monitored components can also suffer undetectable failures. An example would be a breakdown of insulation in a sensor which does not short under normal operating conditions, but causes failure of the sensor in the steam environment of a true demand. The shock which causes the transition to the failed state occurs during the standby period, but it is a failed state only under the highly stressed conditions of a true demand.

3.2.3 FAILURE RATE SHAPE FACTOR

The Failure Rate Shape Factor, β , specifies how the instantaneous conditional failure rate (hazard rate) changes with time. When β is equal to one the failure rate is constant and independent of time. For values of β less than one, the failure rate decreases with time. For values greater than one it increases.

The shape factor models the fact that a component's susceptibility to standby failure mechanisms can change with the past standby service life of the component. For example, the specification of $\beta=2$ implies that the failure rate of that component is increasing linearly with time. The implication is that environmental factors acting on the component during

standby are gradually degrading the component's resistance to failure causes which can transfer it into a failed state. For example, gradual buildup of corrosion products might be considered as an accumulation of small shocks which can transfer a valve into a stuck state with increased probability as its exposure to the process fluid increases. This in turn results in a higher probability that the valve will stick when demanded to open. Failures resulting from this mechanism would have a $\beta > 1$. Time dependent failure rates will be discussed in more detail in Section 3.3.2.

Conversely, if $\beta = 1$ the conditional probability of the component failing during its 10,000 th hour of standby (given it has survived until then) is the same as the conditional probability of its failure at any other time during the standby period. By implication, the constant failure rate model assumes that if the random shocks of the standby period do not cause a transition to a failed state, they have no impact what-so-ever on the component. This is called the exponential failure model because the time dependence of the availability (cumulative probability of survival to time t) follows an exponential distribution.

3.2.4 COMPONENT RENEWAL TYPE

In FRANTIC II-MIT a component can be renewed by either

⁵ The default value of $t_0 = 0$ corresponds to the renewal modeling available in FRANTIC II.

periodic testing (which cycles a component through an active phase of operation) or repair. Renewal has the mathematical effect of resetting the generalized Weibull hazard rate back to the offset time, t_0 .⁵ The following renewal input options are available in FRANTIC II-MIT:

New-New (NN) Renewal \equiv Both test and repair reset the hazard rate. The renewal time, t_r , is automatically set equal to the end of the most recent scheduled test period when the component is not failed, and is set equal to the end of the repair time when it is failed. This type of renewal might model a failure mechanism in which two metallic surfaces cold weld during the standby period (exhibiting a $\beta > 1$ hazard rate). When the component is operated the effect of the cold weld is broken, and the cold welding process must begin again.

Old-Old (OO) Renewal. Neither test nor repair reset the hazard rate. The renewal time is kept at $t=0$, the beginning of the FRANTIC II-MIT calculation. The hazard rate follows its time dependence for the entire period of the calculation. This renewal type might be used to model the failure rate of a large component where various parts exhibit wearout due to abrasion and corrosion. Testing has no effect on these mechanisms, and repair of breakdowns can correct only local problems, so the component's overall failure rate gradually increases.

Complete replacement of a large component would probably justify recalculation of unavailability using the newer component's estimated failure rates. This type of repair is beyond

that envisioned for the repair time modeled in the code, since it would require shutdown of the plant.

Old-New (ON) Renewal. Testing has no effect on the hazard rate, but repair resets it. When the component is found failed at a periodic test, the renewal time is set as the time of the end of repair. (For further information, see the FRANTIC II manual.) This component type most often models components that wearout or gradually deteriorate and are replaced when they fail. For example, a circuit board gradually deteriorates in a humid environment. Testing can not reverse the deterioration, but when the circuit board is found to be failed it is replaced with a new one.

Use of Offset Time With Renewal Types

In FRANTIC II-MIT, the user has the option of specifying an offset time, t_o , which is the time used to calculate the hazard rate immediately after renewal. Its use in conjunction with an OO renewal option will permit the user to start a calculation late in the life of a safety system and initialize the hazard functions of the components so that wearout effects would have already accumulated at the beginning of the calculation. It also gives him the flexibility to initialize wearout of different components at different times. When used with the NN and ON options, it gives the user flexibility in selecting the shape of the time dependence of the failure rate.

3.2.5 TEST CAUSED CHANGES TO SCALE FACTOR

In FRANTIC II-MIT the scale factor for detectable standby

failures may be changed by a user specified factor to reflect the possibility of test caused wearout. See Section 3.5.3 for a discussion of this factor.

The scale factor for undetectable standby failures is not changed by periodic tests in FRANTIC II-MIT. This change was made because it was judged that tests which cannot detect failure mechanisms can not affect them sufficiently in other ways that can increase the components susceptibility to them. See Section 3.5.3 for a more complete discussion of test caused changes in failure rates.

3.2.6 UNAVAILABILITY DUE TO STANDBY FAILURES

Unavailability due to standby failures depends on whether the component is monitored or periodically tested.

Monitored Components

The equations for monitored components are those of FRANTIC II. It uses the asymptotic equations:

Repair to "Good as New"

$$q_m(t) = q_m = \frac{\lambda^{1/\beta} T_R}{\Gamma(\frac{1}{\beta} + 1) + \lambda^{1/\beta} T_R} \quad (3.6)$$

Repair to "Good as Old"

$$q_m(t) = \lambda t^{\beta} - \lambda(t - T_R)^{\beta} \quad (3.7)$$

Where:

T_R - Average repair time

Inspection of these equations shows that the unavailability contribution depends heavily on T_R , the time required to complete repair. If the component can be repaired quickly it will have a small unavailability despite a high failure rate. Equation (3.6) indicates that even with time varying hazard rates the average unavailability of monitored components will approach a constant value when repair resets the hazard rate. If the hazard rate continues to change even after repair, the unavailability equation varies with time, as shown in Equation (3.7).

Periodically Tested Components

In periodically tested components, standby failures can produce three different types of unavailability contributions:

- 1) They can occur but remain undetected during the standby period.
- 2) They can be revealed during a test period.
- 3) They can keep the component down until repair is completed.

During the standby period the unavailability is given by:

$$q(t) = 1 - \exp\{-\lambda[(t+t_o-t_r)^\beta - (t_w+t_o-t_r)^\beta]\} \quad (3.8)$$

The instantaneous unavailability increases as the time since the component was last known to be operational, t_w , increases, because the time during which the failure can occur increases. After a periodic test, the current time becomes the

time the component was last known to be working, so the undetected unavailability returns to zero. Equation (3.8) is derived and discussed in more detail in Section 3.6.2.

At the beginning of a periodic test the total undetected unavailability is Equation (3.8) applied over the entire prior standby interval. The test reveals the failure, but the component remains down until repairs can be made. Therefore, the contribution of standby failures to the component's unavailability remains at ~~at~~ that level until the end of the repair period.

The unavailability resulting from the use of λ_u is similar to equation (5.9), but the component is assumed to be last known working at the beginning of the calculation for the NN⁶ and OO renewal options and at the last renewal time, t_r , for the ON renewal option. The resulting equation is:

$$q_u(t) = 1 - \exp\{-\lambda_u [(t + t_o - t_r)^\beta - (t_o)^\beta]\} \quad (3.9)$$

Where:

For OO and NN component renewal, $t_r = 0$.

For ON component renewal, t_r is set with the probability that a detectable failure occurs.

⁶ The original version of FRANTIC II assumed that every renewal revealed undetectable standby failures. Consequently, for the NN option, where renewal occurs at every test, there was no difference between detectable and undetectable failures.

3.3 DEMAND FAILURE RATE

3.3.1 ENGINEERING INTERPRETATION

The demand failure rate, q_d , models failure mechanisms which are independent of the time of the true demand. It can be input with either monitored or periodically tested components and contributes that constant value to the unavailability calculated for that component at every time point. Although the demand failure rate is a constant, a modification has been made in FRANTIC II-MIT to multiply it by a factor which can change as a function of periodic tests to model test caused changes in the susceptibility of the component to the mechanisms that the parameter represents.

The demand failure rate is used to model failure mechanisms other than those which occur during the standby period. It can represent the probability of:

- Conditions at the time of the true demand that defeat the component's ability to perform its function. An example would be the probability that an electric motor is flooded by a particular Loss of Coolant Accident, and therefore shorts and can not function.
- Failure mechanisms caused by transitions between states. For example, the accelerations of starting and stopping may be the cause of failures during demand, or too much close force during the previous closure may jam the valve and consequently prevent its opening.

- Errors during test and maintenance that leave an originally operable component in an undetected failed condition. This is called a Type A human error by McWilliams and Martz. [Mcw80] An example of this is leaving the Auxillary Feedwater System valved out of the secondary system following a periodic test.

When applying this parameter to periodically tested components one should remember that a demand to operate implies two transitions, both of which can cause failures. A previous transition to standby could have left the component in an undetected failed state, and the current demand to operate can also cause the failure. (Operator error and jammed valves due to closure are examples of failures occurring during transitions to the standby condition.)

3.3.2 SPECIAL USES

Because the demand failure rate is a convenient way to input an unavailbility that remains constant throughout the calculation, it can be used in conjunction with other parameters to produce more flexible modeling of time dependent system unavailability. The following sections present some applications, but are in no way a complete listing of the possibilities.

Monitored Failures in Periodically Tested Components

Some periodically tested components may have some portion of their function monitored while they are standby. For example, the voltage across electrically operated machinery may be

constantly monitored, or idle equipment may be subjected to visual inspections which can reveal some failure mechanisms. Other failure mechanisms may be revealed only by operating the equipment. If the monitored failures are assumed to have a constant failure rate or the component has NN renewal, the average monitored unavailability is asymptotically constant. One could then calculate the average using Equation (3.7) and input it directly using q_d .

If the monitored hazard rate is increasing, one could use q_d in conjunction with the undetectible scale parameter, λ_μ , to model an average unavailability due to monitored components which rises gradually throughout the calculation.

Common Cause Failures

When more than one component can be made ineffective because of the conditions of the true demand, the resultant common cause failure can be modeled by a separate failure event in series with whatever failure event those components affect. Since the true demand is assumed to occur randomly, it can be modeled by a constant unavailability represented by q_d . In this case q_d would equal the fraction of true demands for which the components fail to perform their function.

3.4 TIMES ASSOCIATED WITH PERIODIC TESTING

3.4.1 PERIODIC INSPECTION INTERVAL

The Periodic Inspection Interval, T_2 , sets the time

between the start of successive periodic tests. It is input in days , but is automatically rounded off to an integer number of hours by the code. Most practical applications will involve using a whole number of days as input for this parameter.

3.4.2 FIRST PERIODIC INSPECTION INTERVAL

The First Periodic Inspection Interval, T_1 , allows the user to stagger the periodic testing of various components to reflect the sequence and interval spacing in which tests are actually accomplished. Because the code begins its calculations assuming all time dependent unavailabilities are zero at time zero, system unavailability near the beginning of a calculation may not reflect actual unavailability. First interval effects can be minimized by averaging over many inspection intervals and not selecting instantaneous system unavailabilities from times close to the beginning of the calculation.

3.4.3 SCHEDULED TEST AND MAINTENANCE PERIOD

The Scheduled Test and Maintenance Period, τ , represents the average duration of scheduled periodic testing and maintenance and is input in units of hours. This includes the actual testing time for which the component is unavailable and the time for repairs of the component that one would expect to do on a regular basis to prevent safety related failures later. It does not include unexpected failures which require additional time for repair. For example, consider a component that is inspected every month and found to be able to accomplish its

safety function. However, minor problems are discovered which, if not corrected, could result in a failure of the component at some time in the future. The plant policy is to make minor repairs and repeat the operational test as a matter of course. Since the the component was not in a failed condition at the beginning of the test and the repair was not unexpected, this maintenance policy should be accounted for in the specification of τ

3.4.4 UNSCHEDULED REPAIR TIME

The Unscheduled Repair Time, T_R , accounts for repair that is accomplished when a component is found to be failed, either by monitoring or by periodic testing. The unscheduled repair time accounts for the total time from the discovery of the fault through retesting and qualification of the component for standby service. It does not include normal maintenance that is done on a component, which is accounted for in τ .

During unscheduled repair a component is assumed to be totally unavailable. The (unconditional) unavailability calculated by the code during T_R is the probability that the component requires repair (because it failed) times one. Therefore, the user should account for partial availability by shortening his estimate of T_R . For example, if on line repair of a component takes on the average of 10 hours, during which the component is not available to accomplish its safety function. Requalification for standby takes an additional 4 hours,

during which the component is only 25% unavailable. The average unscheduled repair time should be calculated as:

$$T_R = 10 + .25(4) = 11 \text{ hours} \quad (3.10)$$

3.5 EFFECTS OF IMPERFECT TESTING

The parameters discussed below are directly associated with testing. However, failures during testing which also have a large impact of system unavailability are modeled by q_d . These, of course, refer to the probability of leaving a component in an undetected failed state following the completion of the test period. See Section 3.3.1.

3.5.1 PROBABILITY OF TEST CAUSED FAILURE

The Probability of Test Caused Failure, P_f , represents the probability of failures occurring during periodic tests that would not cause the component to fail to perform its function in the event of a true demand at any time, but which generate the requirement for an unscheduled repair following a normal periodic test. This includes repairs to prevent leakage and contamination, or repairs to correct precursor faults which currently do not interfere with the functioning of the component, but if left uncorrected could lead to a safety related failure in the future. It also includes failures generated by the conditions of the test which do not occur during an accident.

Since a test cycles some components to an operating mode, with its potentially much higher failure rate, a component can fail due to active mode failures which are not related the standby period during a test. The parameter P_f also accounts for these types of failures.

Test caused failures are assumed to be immediately detected, but the component becomes unavailable for the additional unscheduled repair time discussed above. This parameter increases the unavailability of the component by P_f during the test and repair periods, τ and T_R . The component returns to an available status at the end of the unscheduled repair period.

An example of test caused failure is a pump which blows a seal during a flowrate test. The pump could have completed its mission with the blown seal had this been a true demand. However, to prevent excessive contamination and further damage to the pump, repair must be affected, and the pump is assumed to be not available to accomplish its safety function for both τ and T_R .

3.5.2 UNAVAILABILITY TO OVERRIDE TEST AND MAINTENANCE

Unavailability to Override Test and Maintenance represents the probability that a good component can not be used for its intended function should a true demand occur while it is undergoing periodic test and maintenance. It models the fact that periodic tests often require some reconfiguration from the component's standby "ready" mode. It also accounts for the

fact that maintenance and minor repair might make the component unavailable for some fraction of the test and maintenance time.

This parameter should be estimated considering all of the activities that could go on during a scheduled test and maintenance period, τ . It is actually the fractional down time of the component, averaged over τ that is not caused by failures which fall into the category of P_f or Q_n . It is derived from an assessment of both the test procedure and the maintenance activities which normally occur during τ .

3.5.3 TEST CAUSED FAILURE RATE CHANGE FACTORS

Periodic operational tests require that components cycle to an operating configuration. As the component starts, operates, and shuts down, it experiences a series of stresses which may cause changes in its probability of failure when it is returned to standby. In FRANTIC II-MIT this effect is modeled by factors which multiply the current value of the standby failure rate scale factor, λ_o , and q_d every time a periodic test is accomplished. After n tests the standby failure rate becomes:

$$\lambda_n = (f_\lambda)^n (\lambda_o) \quad (3.11)$$

Where:

λ_o - Value of the initial scale factor input by the user.

f_λ - Standby failure rate test factor.

After n tests the demand failure rate becomes:

$$q_d = (f_t)(f_d)^n(q_{do}) + (1-f_t)q_{do} \quad (3.12)$$

Where:

q_{do} - Value of q_d input by the user.

f_t - Fraction of q_d which is affected by the test.

f_d - Demand failure rate test factor.

The factor f_t allows the user to keep selected portions of q_d constant for the entire period of the calculation. It also allows him more flexibility in selecting how q_d will vary as a function of periodic tests.

The actual value of n used in the code depends on the renewal option.

NN Option: n represents the number of tests since the beginning of the calculation. Note that one can accumulate test caused wearout while also resetting the Weibull hazard rate at each periodic test.

OO Option: n represents the number of periodic tests since the beginning of the calculation or $t=0$. However, if the user has input a positive t_o , the code will calculate n based on a total accumulated time since renewal of t_o+t .

ON Option: n represents the number of periodic tests from the last t_r , where t_r will have values accounting for repair renewal during each periodic test period of the calculation. Each one's contribution to the unavailability will be weighted

according to the probability that the component was failed at that particular periodic test.

The default values of f_λ , f_d and f_t are 1.0, corresponding to no test caused changes. Values of f_λ and f_d greater than 1.0 represent the wearout effects of on-off cycling and active functional periods required by operational tests. Values less than 1.0 represent improvements in equipment and procedures because of lessons learned during the tests.

It is important to remember that f_λ and f_d account for the effect of periodic testing on the probability that the standby failure mechanisms modeled by λ and demand failure mechanisms modeled by q_d will occur after the test is completed. Consequently, the evaluation of f_λ and f_d must consider what $\lambda(t)$ and q_d account for in the first place.⁷

Some examples of test caused changes in failure rates might be:

- 1) An operational test produces highly stressed cyclic loading on a valve stem, creating fatigue cracks. The probability that the valve will fail while closing during an accident gradually

⁷ Periodic testing can also change the operating failure rates of the components. This involves an entirely separate set of failure parameters, which would have to be coupled to the unavailability calculation using a phased mission analysis to obtain the total unreliability of the system for performing a mission of a given duration. The treatment of phased missions is not the subject of this thesis.

increases as the periodic tests are repeated. This would be modeled by using a value greater than one for f_d .

2) Cycling to the active phase increases the seepage rate of water from the High Pressure Coolant Injection System. The increased flow of water enhances the rate of failure mechanisms caused by water soakage during standby. This would be modeled by using a value greater than one for f_λ .

3) Operational tests reveal ways to correct potential failure mechanisms. This would be modeled by using a value less than one for f_λ or f_d depending on the types of failure mechanisms being eliminated. (Note, however, that one would not account for one time corrections using these parameters, since the factors model gradual changes to failure rates.)

4) Operational tests reveal areas for improvement in operator training and procedures. They also provide valuable experience for the crew running the plant. This gradually reduces the probability of operator error. It would be accounted for by using a value less than one for f_d .

3.5.4 TEST ERROR CARRYOVER FACTOR

The input parameter C_f accounts for the nondetection of component failures which should be detected by a periodic inspection, but are not because of human error. This type of error is called a Type B error by McWilliams and Martz.

[Mcw80] With humans monitoring or accomplishing the test, there is a probability, C_f , that a detectable failure that existed just prior to the test is not found, and the component

remains in a failed state throughout the next standby period. Consequently, there is an additional contributor to unavailability, $C_f Q_n$, in the next standby period. For example, while verifying the operation of valves, an operator cycles one valve twice, leaving a second unchecked. If the second valve is failed, it will remain so throughout the next standby period.

3.6 GENERALIZED WEIBULL HAZARD RATE

In this section the concepts and procedures necessary to use the generalized Weibull hazard rate to model changes in standby failure rates will be developed ⁱⁿ more detail. First, some of the physical processes which can produce changes in a component's failure rate are described. Physical processes which occur during testing are more appropriately modeled by the test change factors discussed in Section 3.5.3. Next, a generalized formulation of the Weibull hazard rate is introduced to model the standby failure rate of a component. It is shown that a large variety of smoothly varying time dependencies can be modeled with this mathematical formulation.

Before continuing, it should be emphasized that the generalized Weibull hazard rate is an extremely convenient way to represent conditional failure rates which vary with time. All of FRANTIC II-MIT's calculations are based on using the conditional failure rate (hazard rate) to obtain instantaneous component unavailability by direct integration. To use the code, one must first know the time dependence of the conditional

failure rates of all the components. It is not the purpose of this thesis to suggest ways to functionally determine the hazard rates. Techniques are available in the literature. (e.g. Many methods are based on life testing in which the observed times to failure may be assumed to follow a Weibull Distribution Function. [Br73, EP1558, He81, McC81]) However, these estimation methods should not be confused with the stochastic process whose time dependence happens to follow one of the curves the generalized Weibull hazard rate can generate.

With the exception of the method for accounting for test caused changes in the standby failure rate, the models in FRANTIC II-MIT are more comprehensive than those of FRANTIC II. As this section demonstrates, the two parameter Weibull Hazard Function used in the original code does not give the analyst much flexibility for representing time dependence. Also, the original version does not permit any variation in the demand failure rate as a result of testing. These limitations are overcome by the models used in FRANTIC II-MIT

Methods for statistically estimating all the parameters available in FRANTIC II-MIT is beyond the scope of this thesis. It would seem reasonable to expect that a Bayesian approach for combining knowledge gained in analyzing the root causes of failures with the statistical data obtained from failure frequencies attributed to known and unknown failure mechanisms could be useful in the estimation process. This is discussed in the section on recommendations for further research.

3.6.1 SUSCEPTIBILITY OF COMPONENT FAILURE MECHANISMS

The instantaneous conditional failure rate can be thought of as a measure of a component's susceptibility at a specific time to the physical mechanisms that cause it to fail. The susceptibility depends upon the accumulated effects of the component's prior operational and exposure history. In some cases it might also model the effects of an environment which is changing as a predictable function of time.

A constant failure rate model carries the assumption that the susceptibility of the component to failure is not changed by its previous standby and operating history. When we use a constant failure rate model we are saying that a component that has been in service for a period of time is no more susceptible to failure than one that has been just put into service. The environment to which the component has been exposed and its operational history have produced no degrading effects. For many types of components this assumption may be reasonable. Periodic maintenance and replacement of worn parts in complex components can combine to keep the susceptibility constant. In fact this assumption has been made in most nuclear safety applications to date.

In FRANTIC II-MIT the instantaneous failure rate is modeled with a time dependent hazard function, $\lambda(t)$. It represents a conditional failure probability. Given the component experienced the physical processes modeled by the hazard function during the interval prior to time t and is not failed at

time t , the probability that it will fail in the interval $(t, t+dt)$ is $\lambda(t)dt$.

In choosing a representation for $\lambda(t)$, the analyst should have an understanding of the physical processes which can either improve or degrade the survivability of the component. If they occur during the standby period, $\lambda(t)$ will gradually increase or decrease, depending on whether a wearout or burn-in effect is produced. If they occur during a periodic test, a discontinuous jump in $\lambda(t)$ will result.

Susceptibility to failure can come from both internal flaws and environmental stresses. Factors which might cause a component's failure rate to increase with time include, for example:

- Operational wear of surfaces or parts of the component which affect its tightness of fit, smoothness of operation, etc.
- Wear or corrosion produced by the flow of a process fluid over, around, or through a component.
- Fatigue due to on-off cycling, vibration, temperature changes, etc.
- Radiation damage or embrittlement.
- Buildup of corrosion products, rust, etc. on components.
- An increasingly stressful environment under which the component must operate. This might be caused by the gradual wearout of other components in a long term situation.

- Accumulated effects of past environmental stresses on the component which have caused a random amount of undetectable or uncorrectible damage to the component.
- Effects of prolonged idleness, including cold welding, drainage of lubricant, discharge of batteries, etc.
- Human error on the part of operations or maintenance personnel which allows precursors of failure mechanisms to go undetected.

Factors which could decrease the failure rate include, for example,

- Correction of design or manufacturing errors.
- Improvement in operating procedures or operator proficiency which results in less stress on the component.
- Redesign, reconfiguration or improvement or other system components which result in a less stressful operating environment for the component of interest.
- Replacement of a failed component with a new component.

For components which are normally idle during standby, the test period usually requires cycling to the active mode, so there are two distinct phases during which $\lambda(t)$ can change. During standby, it is reasonable to assume that an idle component will transfer into the failed state due to the influences of inherent mechanisms or external shocks. These effects can not normally be correlated to any particular event. Although not strong enough to cause immediate failure, some of these influences may have a degrading effect on the component. Con-

sequently, the susceptibility to actual failure upon reoccurrence of a similar stress may be increased. In FRANTIC II-MIT this gradual buildup of susceptibility is modeled by the generalized Weibull hazard rate formulation of the standby failure rate.

During periodic test and maintenance periods, the component can be subjected to a variety of influences. As it cycles through an active phase, it will be subjected to the shock of starting and stopping as well as the wear induced by active operation. Maintenance actions can detect and correct problems, reducing susceptibility to failure. However, maintenance errors might induce additional stresses in the component and inadvertently increase its failure probability. Parts or entire components may be replaced during repair. As a result, the susceptibility of the component after a periodic test might be quite different from what it was at the beginning of the test. In FRANTIC II-MIT the discontinuous change in failure probability produced by a periodic test can be modeled by:

- 1) Resetting the Weibull hazard function time to the offset time.
- 2) Multiplying the Weibull hazard function scale factor, λ , by a user input factor at each test.
- 3) Multiplying the Constant Demand Failure Rate by a user input factor at each test.

3.6.2 EFFECTS OF OFFSET TIME ON HAZARD RATE

As part of this work, FRANTIC II-MIT has been modified to represent the conditional failure rate by a generalized form of the Weibull hazard rate (originally presented as Equation (3.5) in Section 3.2).

$$\lambda(t) = \beta \lambda_n (t+t_o-t_r)^{\beta-1} \quad (3.13)$$

Figure 3.4 shows the the effect of the offset time and its relationship to the calculation time and the renewal time. At the begining of a calculation both t and $t_r = 0$, and the time variable for the hazard function equals t_o . The combination of λ_o , β and t_o sets the value of hazard function at this time. As the calculation time increases, the value of the hazard rate changes dependent on t_o and β , with t_o playing an important part, because it affects the factor by which the time variable changes as the calculation proceeds. When renewal occurs t_w is updated from zero to the calculation time, then immediately after renewal the time variable is again at t_o . So the effect of renewal in FRANTIC II-MIT is to reset the hazard function back to the the value it had at the begining of the calculation and start its time variance over again.

Using the generalized Weibull formulation of the hazard rate, the differential change in a component's unavailability due to undetected failures during standby because of failures between t and $t+dt$ can be expressed as:

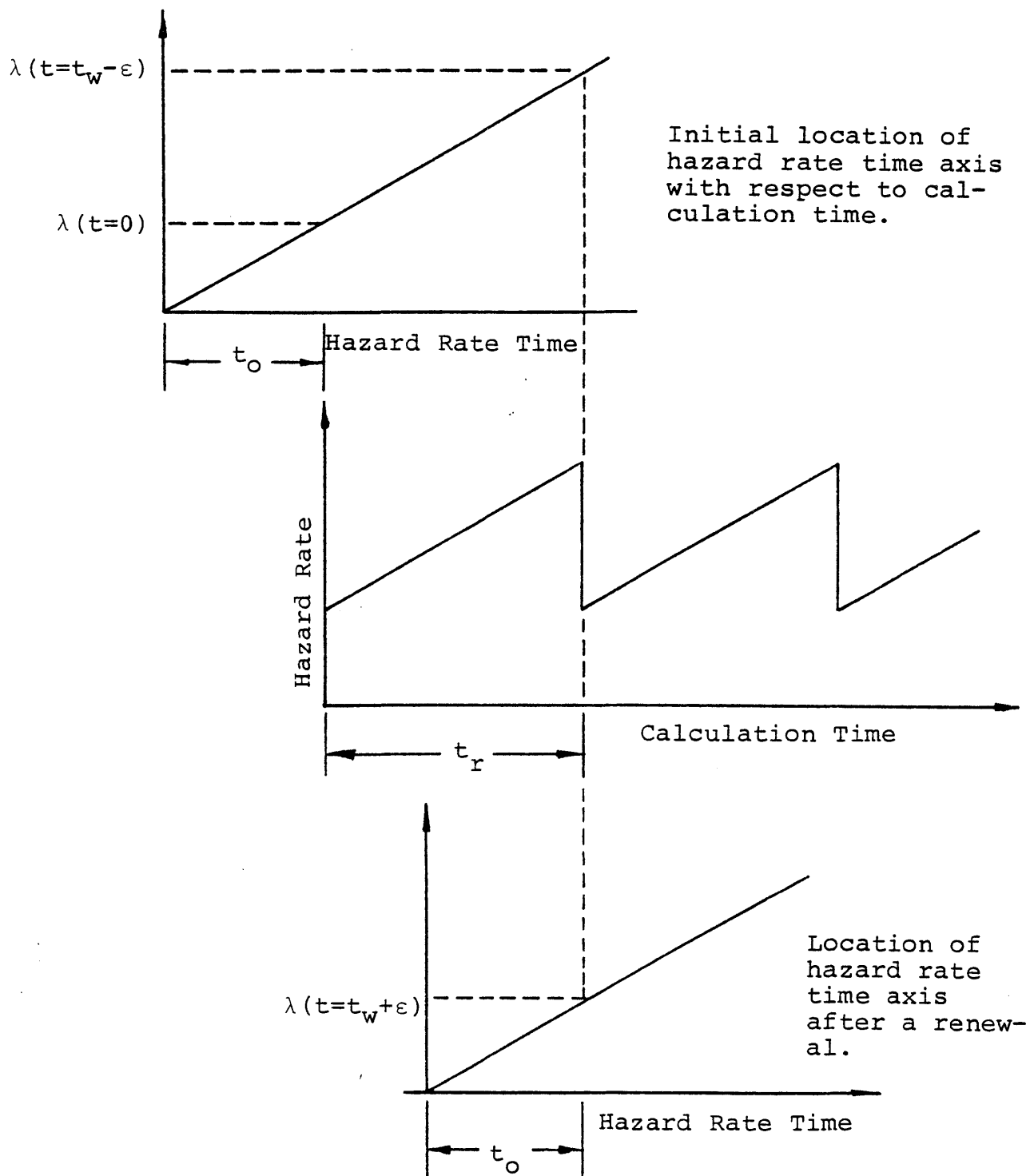


Figure 3.4. Use of Offset Time and Renewal Options to Obtain Time Dependent Failure Rates With a Generalized Weibull Hazard Rate.

$$d[q(t_w, t)] = [1 - q(t_w, t)] \beta \lambda (t + t_o - t_r)^{\beta - 1} dt \quad (3.14)$$

Where:

$q_\lambda(t_w, t)$ - Unavailability due to undetected standby failures which have occurred since the last time the component was known to be working.

t_w - The last time the component was known to be working, either at the end of the last test period or the end of the last repair period.

This formulation is the same as for the reliability of an unreparable component, since the transitions to the failed state during standby can not be repaired until they are detected by the next periodic test. The term $[1 - q(t_w, t)]$ represents the fact that in order to fail between t and $t + dt$, the component must first survive to time t . When the component is known to be operational, say at the end of test or repair, $t = t_w$, and $q_\lambda(t_w, t_w) = 0$.

If this equation is integrated from the time a component was last known to be working to time t , one obtains:

$$q(t_w, t) = 1 - \exp\{-\lambda[(t + t_o - t_r)^\beta - (t_w + t_o - t_r)^\beta]\} \quad (3.15)$$

This is the expression for unavailability due to unrepaired standby failures in the interval between t_w and t which is implemented in the FRANTIC II-MIT code. Because of its conven-

ient analytical form, it is quickly evaluated by the computer and provides an efficient vehicle for modeling a large variety of time dependent hazard rates.

3.6.3 ADVANTAGES OVER FRANTIC II HAZARD RATE

The use of the offset time in the Weibull hazard rate provides much flexibility in establishing the shape of the instantaneous failure rate, while not increasing the complexity of the unavailability calculation. With the offset time available as an input parameter to the FRANTIC II-MIT code, one has the option to set the initial time from which the failure rate will be integrated to any point on the time axis of the Weibull hazard rate time axis. Since the hazard rate is just a mathematical function, there is no physical reason to insist that renewal begin at time zero. In fact, a renewal process which resets the Weibull hazard function to only time zero restricts the time variation of the resultant failure rate function considerably. This can be made more clear by the following example.

Figure 3.5 plots five failure rates modeled with the two parameter Weibull hazard rate available in FRANTIC II. The five curves all have a value of zero at time zero and rise to $1.0E-5$ per hour after 20 years. These curves correspond to the values of β and λ given in Table 3.1.

These curves illustrate the problem which arises from using only two parameters to establish the hazard rate. Say the analyst is modeling a component with an Old=Old (OO)

renewal option ($t_r=0$ for the entire calculation). The data indicate that the failure rate of the component rises by a factor of ten from $1.0E-6/hr$ to $1.0E-5/hr$ over a twenty year period. To determine the shape factor, β , he divides the value of the hazard rate at $t=20$ years by its initial value, say at 1 hour.

$$\frac{\lambda(t_1)}{\lambda(t_0)} = 10 = \frac{\beta\lambda_0(174,200)^{\beta-1}}{\beta\lambda_0(1 \text{ hour})} \quad (3.16)$$

When this equation is solved, he obtains a value for the shape factor of $\beta = 1.19$. Then knowing β and the value of the hazard rate at one hour, he calculates the scale factor from the expression for the hazard rate:

$$\lambda_0 = \frac{\lambda(t)}{\beta} = \frac{10^{-6}/hr}{1.19} \quad (3.17)$$

| Curve | β | λ_0 | Time when $\lambda(t) = 0.1x\lambda(20 \text{ years})$ |
|-------|---------|----------------|--|
| 1 | 1.19 | $8.4x10^{-7}$ | 1 hour |
| 2 | 1.26 | $3.5x10^{-7}$ | 1 day |
| 3 | 1.42 | $4.5x10^{-8}$ | 1 month |
| 4 | 1.77 | $5.3x10^{-10}$ | 1 year |
| 5 | 2.00 | $2.9x10^{-11}$ | 2 years |
| 6 | 2.66 | $7.3x10^{-15}$ | 5 years |

Table 3.1. Parameters used for plotting curves in Figure 3.5. Values of β and λ_0 are chosen so that $\lambda(t)$ reaches 0.1 times the 20 year values at the times shown.

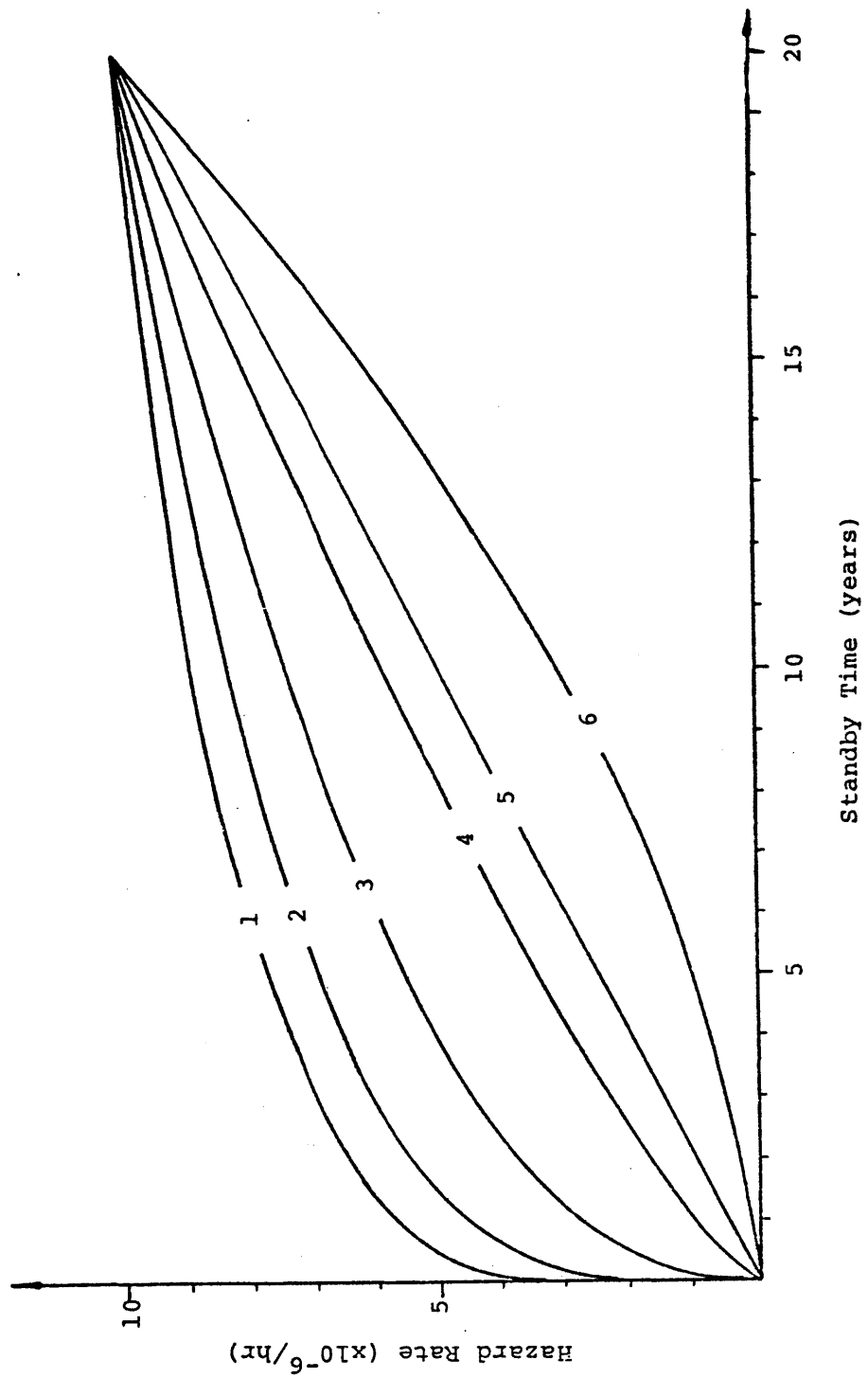


Figure 3.5. Five Failure Rates Modeled With a Two Parameter Weibull Hazard Rate Which Rise From 0 at Time Zero to $1.0 \times E^{-5}$ per hr at 20 Years.

The resulting scale factor is $\lambda=8.4E-7$.⁸ When the hazard rate is plotted versus time, the top curve in Figure 3.5 results. Note that this curve rises to 50 percent of its 20 year value by 6 months, and 80 percent after only 6 years. For the remaining 14 years it is a good approximation of a constant failure rate, since it changes by only 20 percent. (This should not be unexpected, since beta is close to 1.0.) This curve may not be a good representation of the expected increase of component susceptibility to failure over the 20 year period. The curve's shape indicates that the influences which change the component's susceptibility to failure have four times more impact during the first six years than during the last fourteen.

A more gradual rise throughout the entire period may be a better representation of component degradation. For example, external environmental stresses would be expected to occur randomly throughout the component's life and should produce a relatively constant rate of rise in the component's susceptibility to failure. In fact, a hazard function which curves upward during the latter part of the component's life might be more physically reasonable. For example, it might apply to components in which some threshold or margin of safety must be exceeded, after which susceptibility increases at an accelerating rate.

⁸ Note that the scale factor's units depend on the value of $(\beta-1)$.

The remaining curves in Figure 3.5 were derived by making the time at which the failure rate reaches one tenth of its 20 year value progressively longer. It is evident that the hazard function rises more gradually as the 'one tenth' time increases. However, these curves have the disadvantage of producing very low failure rates during the early part of the interval. For example, the use of one year as the 'one tenth' time results in failure probabilities of less than $1.0E-7/\text{hour}$ during the first six months. The use of 5 years, which provides an upward curving failure rate as a component's age increases, produces very small failure rates during the first three years.

Both a gradual rise in the failure rate and physically reasonable values during the first portion of a time interval can be obtained by use of the offset time, t_0 . With the offset time one can select the portion of the Weibull hazard function curve that best matches the actual time dependence of the failure rate. Figure 3.6 illustrates this concept. It again illustrates a situation in which one desires to model a failure rate which rises by a factor of ten over a twenty year period. This time, however, the generalized Weibull hazard rate is available. The 20 year interval does not have to begin at time zero. It may now be any 20 year interval on the hazard rate curve starting at the offset time. The only requirement is that real time elapse a total of 20 years while the hazard rate rises from $1.0E-6/\text{hr}$ to $1.0E-5/\text{hr}$.

For comparison with Figure 3.5, Figure 3.6 has been constructed for offset times of 1 hour, 1 day, 1 month, 1 year, and 5 years. In Figure 3.6 the Weibull hazard function varies from that calculated at the offset time to that calculated for 20 years plus the offset time as t varies from 0 to 20 years. For this example the value of t_0 has already been established. Beta can be found by taking the ratio:

$$\frac{\lambda(20 \text{ years})}{\lambda(0)} = \frac{(175,200 + t_0)^{\beta-1}}{t_0^{\beta-1}} \quad (3.18)$$

The parameter λ can then be determined knowing that $\lambda(t+t_0)=1.0E-5/\text{hr}$. The values of β and λ that were used to plot the curves in Figure 3.6 are given in Table 3.2.

| Curve | β | λ_0 | Offset Time |
|-------|---------|-----------------------|-------------|
| 1 | 1.19 | 8.4×10^{-7} | 1 hour |
| 2 | 1.26 | 3.5×10^{-7} | 1 day |
| 3 | 1.42 | 4.5×10^{-8} | 1 month |
| 4 | 1.76 | 5.9×10^{-10} | 1 year |
| 5 | 2.00 | 2.6×10^{-11} | 2.2 years |
| 6 | 2.43 | 9.4×10^{-14} | 5 years |
| 7 | 7.48 | 3.7×10^{-44} | 50 years |

Table 3.2. Values of β , λ_0 , and offset time used to obtain the hazard function curves shown in Figure 3.6.

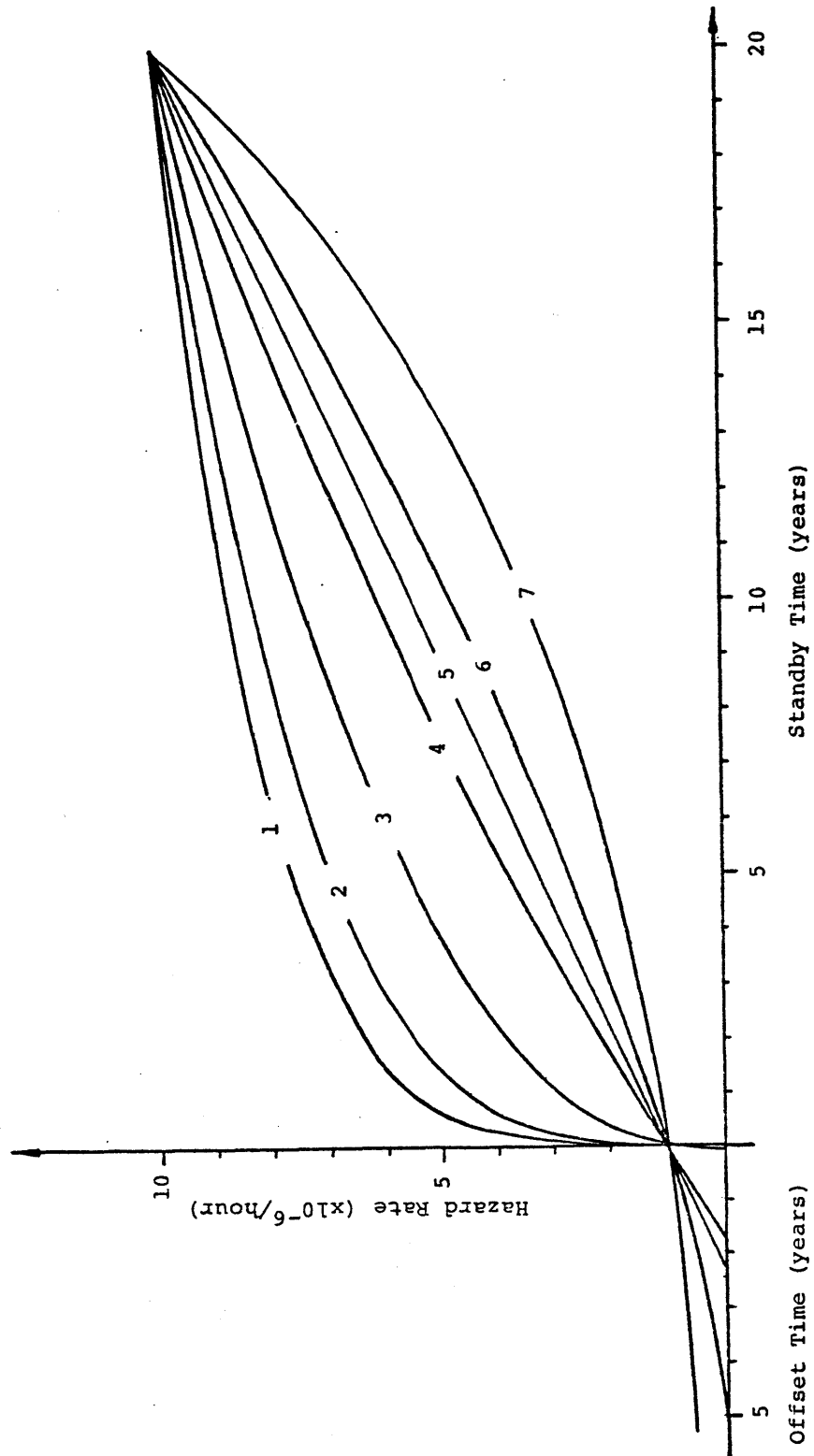


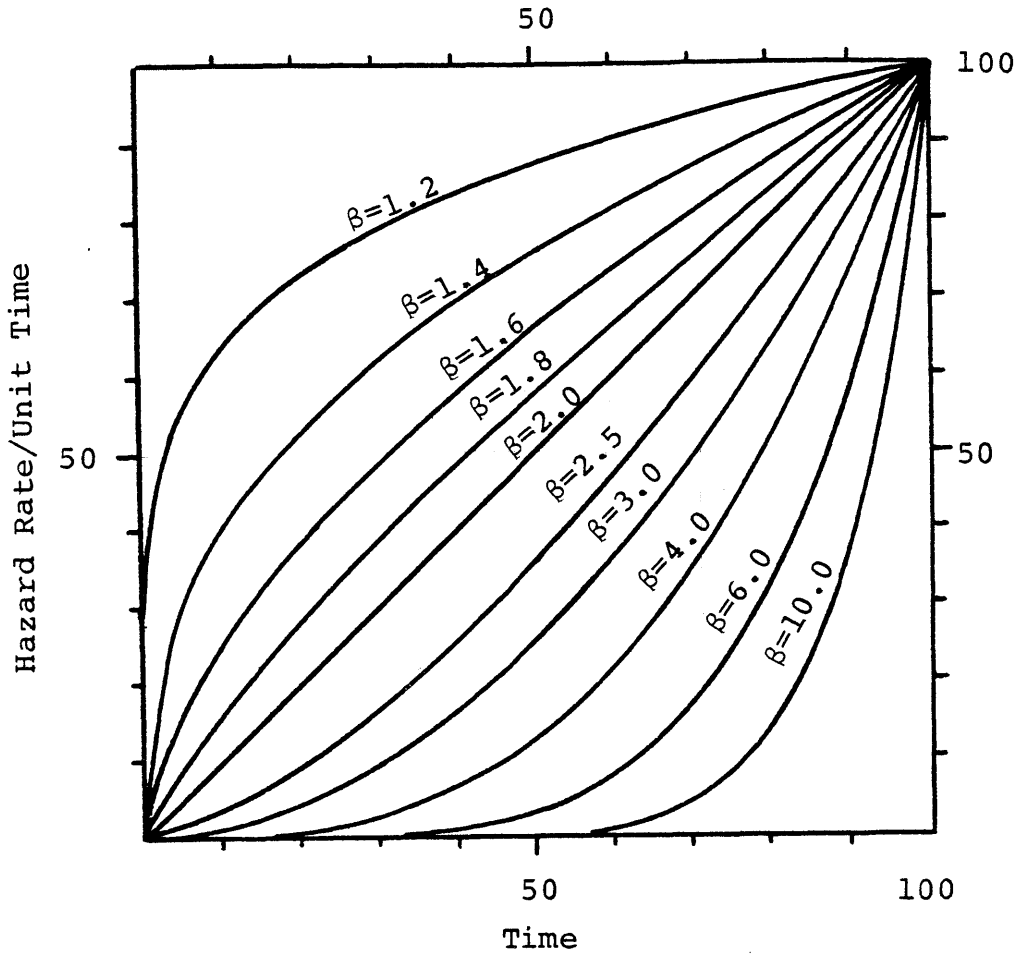
Figure 3.6. Five Failure Rates Modeled With a Generalized Weibull Hazard Rate Which Rise From 1.0E-6/hr at Time Zero to 1.0E-5/hr at 20 Years.

A comparison between Figures 3.5 and 3.6 shows that the use of an offset time allows one to obtain the gradual rise in failure rate modeled by a shape factor of 2 to 3 while not having to accept very small failure probabilities during the first few years of the 20 year period. The offset time has "chopped off" the initial rise from zero and retained the overall shape characteristic of a particular value of beta.

3.6.4 ESTIMATING INPUT PARAMETERS

Use of the generalized Weibull hazard rate implies that the analyst believes that the time dependence of the conditional failure rate follows a power law time dependence. This assumption can be tested using statistical hypothesis testing. The three parameters needed to define the time dependence are λ_0 , β , and t_0 . These three parameters can be estimated by knowing the value of the hazard rate at any three points in time. One simply writes down the equation for the hazard rate at each point of time. With the known values of hazard rate and time substituted into the equations, he now has three equations with three unknowns.

It is also possible to estimate the input parameters graphically. Figure 3.7 is a family of curves derived from the Weibull hazard rate. They are normalized so that each rises from zero to 100 unavailability units in 100 time units. Each curve has a value of beta and lambda associated with it, as shown in the table below the figure.



| β | λ | β | λ |
|---------|-----------|---------|-------------------------|
| 1.2 | 33.18 | 2.5 | 4.000×10^{-2} |
| 1.4 | 11.32 | 3.0 | 3.333×10^{-3} |
| 1.6 | 3.943 | 4.0 | 2.500×10^{-5} |
| 1.8 | 1.395 | 6.0 | 1.667×10^{-9} |
| 2.0 | 0.500 | 10.0 | 1.000×10^{-17} |

Figure 3.7. Normalized Weibull Hazard Rate. The curves rise from 0 to 100 unavailability units in 100 time units (TU).

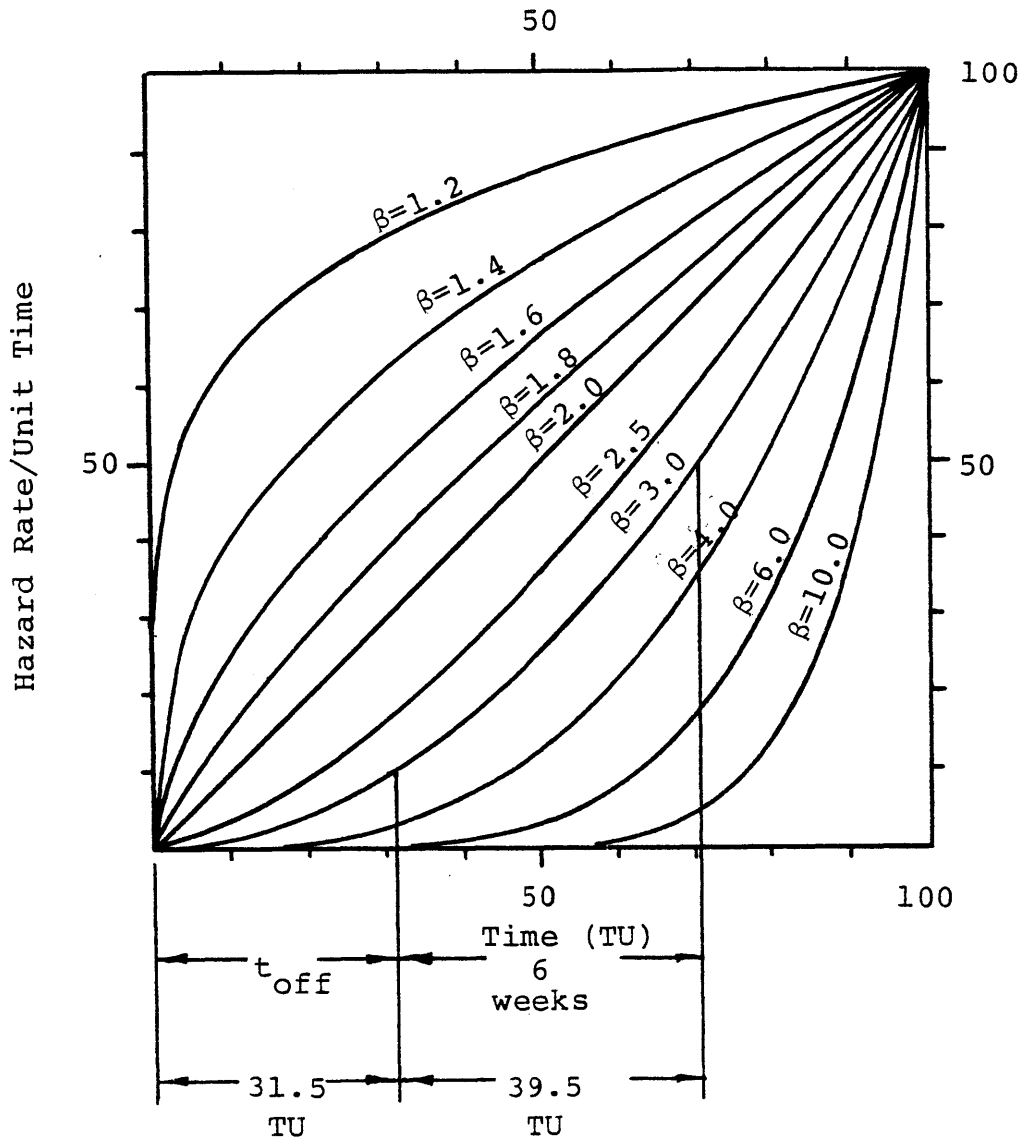
Say the analyst believes the curve associated with $\beta=3$ has the shape that he feels best matches a hazard rate that rises from 1×10^{-6} to 5×10^{-6} /hr over a 6 week period. (This is a New-New failure mechanism.) Figure 3.8 shows how he might use the family of curves to obtain his input for the code. A factor of five increase in the failure rate results when the normalized failure rate rises from 10 to 50. The $\beta=3$ curve crosses a value of 10 at 31.5 time units. It crosses 50 at 71 time units or 1,008 hours. The actual time interval for the rise is six weeks or 1,008 hours. The offset time corresponds to the time from the zero point to the time point of the initial value of the hazard rate. It can be found by taking the ratio,

$$\frac{t_0}{31.5 \text{ TU}} = \frac{1004 \text{ hours}}{40 \text{ TU}} = \frac{1004}{(71-31.5)} = 203.8 \quad (3.19)$$

The arbitrary time unit of 31.5 is the offset time, and it is equivalent to 804 hours. Having obtained t_0 from the graph, one can then calculate λ_0 from the formula for the hazard function, Equation (3.5). At the beginning of the calculation interval $t=t_r=0$, and:

$$\lambda_0 = \frac{\lambda(t)}{\beta(t_0)^{\beta-1}} = \frac{10^{-6}}{3(804)^2} \quad (3.20)$$

When this equation is solved, the result is $\lambda_0 = 5.16 \times 10^{-13}$. The estimates of the input parameters can be checked by calculating the hazard rate at the end of the interval.



From the graph

$$t_{\text{off}} = \frac{31.5 \times 1008 \text{ hours}}{39.5} = 804 \text{ hours}$$

At $t = 0$

$$\lambda_0 = \frac{\lambda(t)}{\beta(t_0+t)^{\beta-1}} = \frac{1 \times 10^{-6}/\text{hr}}{3(804)^2}$$

$$\lambda_0 = 5.16 \times 10^{-13}$$

Figure 3.8. Use of the Normalized Weibull Hazard Rate Curves to Obtain Offset Time.

$$\lambda(t) = \beta \lambda_0 (t_0 + t)^{\beta-1} = 3(5.34 \times 10^{-13}) (1004 + 790)^2 \quad (3.21)$$

The result is $\lambda(t) = 5.08 \text{E-}6/\text{hr}$ after 6 weeks, which is 2% high, but well within the accuracy expected from graphical estimation. It should be noted that failure data usually do not support the requirement for more precise estimation methods.⁹

If the analyst believes that the hazard rate is rising linearly, corresponding to equal weighting of environmental degradation over a period of time, the estimation of t_0 is very easy to accomplish using the straight line on the graph. Say the analyst wants to model a hazard rate which has an initial value of $5.0 \text{E-}6/\text{hr}$ and rises to $7.0 \text{E-}6/\text{hr}$ at the end of two years. This corresponds to a rise from 50 to 70 unavailability units on the graph over a time interval from 50 to 70 time units. Applying the ratio method to determine t_0 ,

$$\frac{t_0}{50 \text{ TU}} = \frac{2 \text{ years}}{20 \text{ TU}} \quad (3.22)$$

*↑ ΔT time units
(50 → 70)*

⁹ Statistical procedures such as maximum likelihood estimation offer more efficient estimation of the parameters provided the third Weibull parameter is permitted to be both positive and negative. These techniques must be used with caution, since the failure data may not all reflect a sampling of the same segment of the hazard rate curve.

yielding $t_0 = 5$ years. The scale factor can then be calculated from the formula for the hazard rate at time zero,

$$\lambda_0 = \frac{\lambda(t_0)}{\beta t_0} = \frac{5.0E-6/\text{hr}}{2(5)(365)(24)} \quad (3.23)$$

yielding $\lambda_0 = 5.71E-11$.

Figure 3.9 shows how the shape of the failure rate curve can depend on the offset time as well as the shape factor, β . The curves start with the failure rate being $1.0E-6/\text{hour}$ at an offset time of either 1, 5, or 10 years. The curves then vary with shape factors of either 1.5, 2.0, or 3.0. Note that the curves appear to become more linear and rise at a slower rate as the offset time increases. This is because the same amount of time produces a smaller factor change in $(t+t_0)$ as the offset time increases.

3.6.5 ALTERNATE MODELS OF TIME DEPENDENCY

An alternate way to obtain failure rates which rise gradually from some initial value is to combine failure mechanisms having different time dependencies with an OR gate. Say, for example, that either random faults (modeled by a constant hazard rate) or a time dependent failure mechanism can cause a particular component to fail. The initial failure rate will depend entirely on the random failure mechanisms. The "wearout" mechanism gradually becomes more probable until it dominates the failure probability. This component can be represented by the Top Event a two "component" system combined

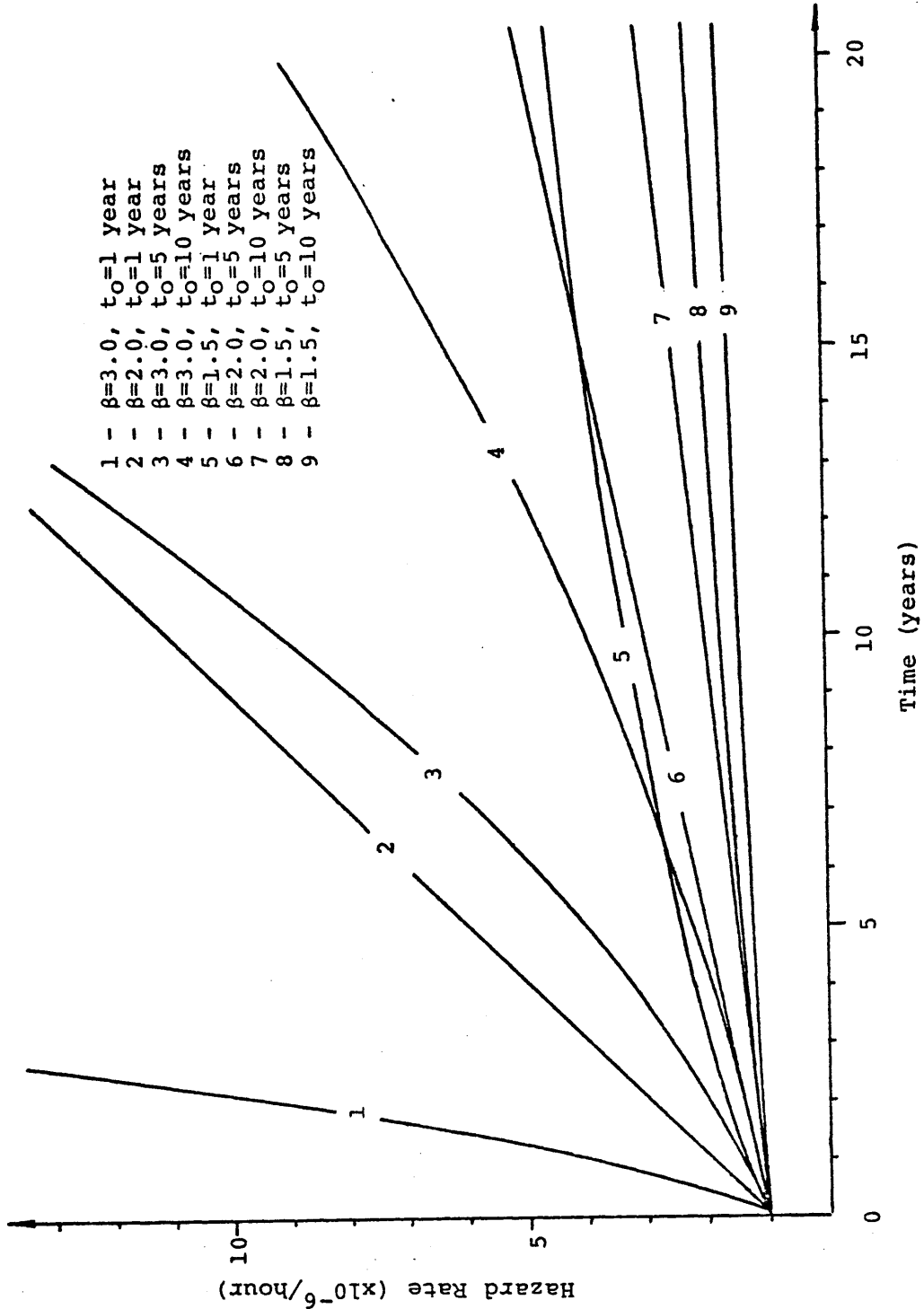


Figure 3.9. Failure Rates for Selected Values of $\beta, \lambda_0,$ and t_0 Which Start at the Same Initial Value

by an OR gate. Figure 3.10 summarizes the individual component failure input parameters and the resulting hazard rate and unavailability as a function of time.

Use of an OR gate can be extended to include a "burn-in" mechanism, which initially has a large value but gradually becomes less probable. The probability of the resultant OR gate would be the classic "bathtub curve."

The use of the OR to explicitly model various types of failure mechanisms has the disadvantage of doubling or tripling the number of minimal cutsets in which the component appears. It also requires breaking down component failure data for input to the computer code only to have it recombined in the Boolean equation. The curves derived using the offset time can be considered to be an approximation of the combination of failure mechanism curves in an efficient form for use in a fault tree.

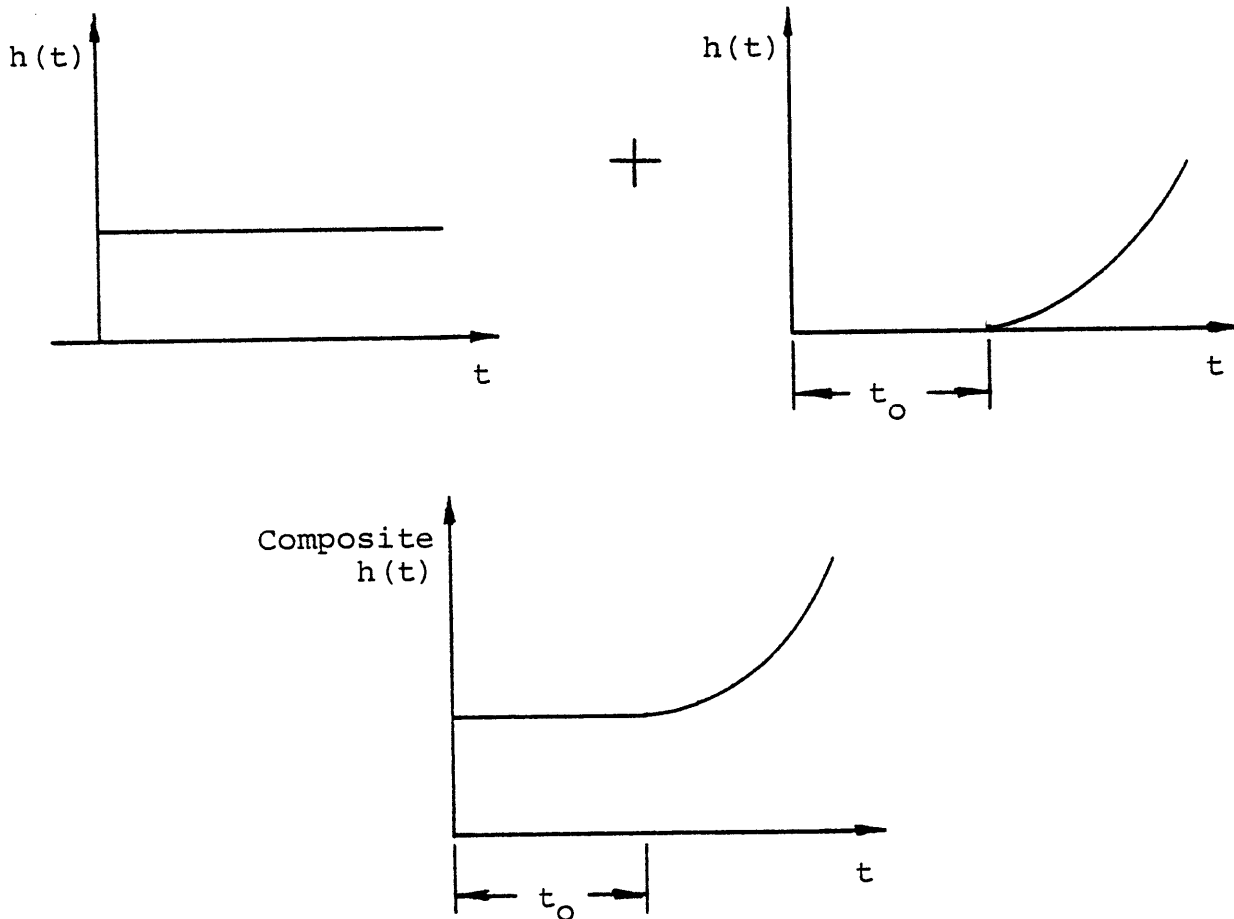
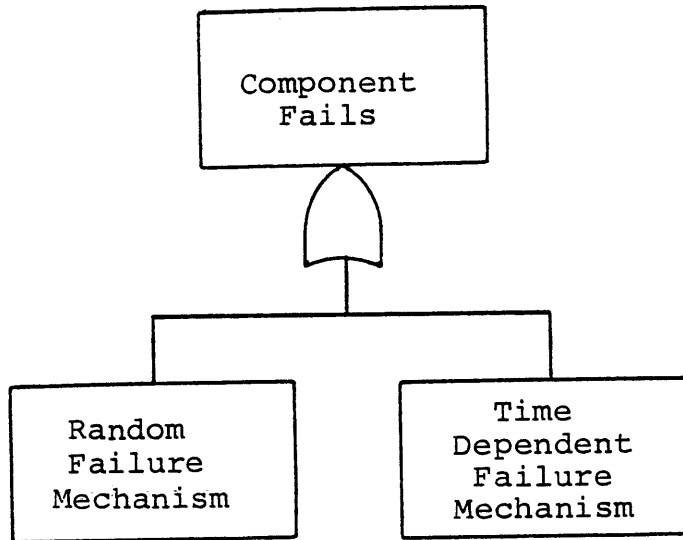


Figure 3.10. Use of an OR Gate to Obtain a Failure Rate Time Dependence Which Does Not Follow a Power Law.

CHAPTER 4

APPLICATION TO SINGLE COMPONENT SYSTEMS

This Chapter discusses the factors which can influence the periodic testing policy of systems that can be represented by a single component failure mechanism. First the effects of demand and standby failures in determining the optimum test interval of a single component system are addressed. Then subroutine OPTEST is introduced. It provides quick estimates of the optimum test interval of single component systems and determines the sensitivity of the system's unavailability to changes in the test interval. It is designed to use data input to FRANTIC II-MIT using the COMPONENTS dataset. OPTEST is used to illustrate the relative importance of the various contributors to a component's unavailability when failure rates are assumed to be constant in time. Finally, the effects of time dependent failure rates and the various renewal options on the selection of test policies are addressed. These include both variations of the standby failure rate with calendar time and test caused changes in the failure rate.

4.1 DEMAND VERSES STANDBY FAILURES

Demand failures (discussed in Section 3.3) and standby failures (discussed in Section 3.2) are the primary causes

of component failure during the standby period, the effects of undetectable and carryover failures being usually a small fraction of the total unavailability. The division of failure mechanisms between the two can have a great effect on the usefulness of periodic testing. If failures are primarily due to standby failures, periodic tests perform the function eliminating failures that occur and lie undetected during the standby period. If the tests are primarily demand related, the tests are just another demand which can cause failure, and they accomplish no purpose from the point of view of the FRANTIC model.¹

4.1.1 RATIO OF OBSERVED DEMAND AND STANDBY FAILURES

Figure 4.1 shows the effect of a change in the ratio of demand to standby failure mechanisms in a periodically tested component. The curves are plotted for the condition that the failure frequency upon test at a 30 day interval is 0.001. The failures are divided into demand and standby failures by the relation (rare event approximation):

$$q_{\lambda}(T_2) + q_d = 0.001 \text{ at } T_2 = 30 \text{ days} \quad (4.1)$$

¹ There are other reasons for accomplishing periodic tests, one of the primary being the prevention of an increase in the failure rate through maintenance. FRANTIC II-MIT does not contain provisions for maintenance strategies other than the New-New renewal option. The Old-New option implies renewal upon failure, which is not maintenance.

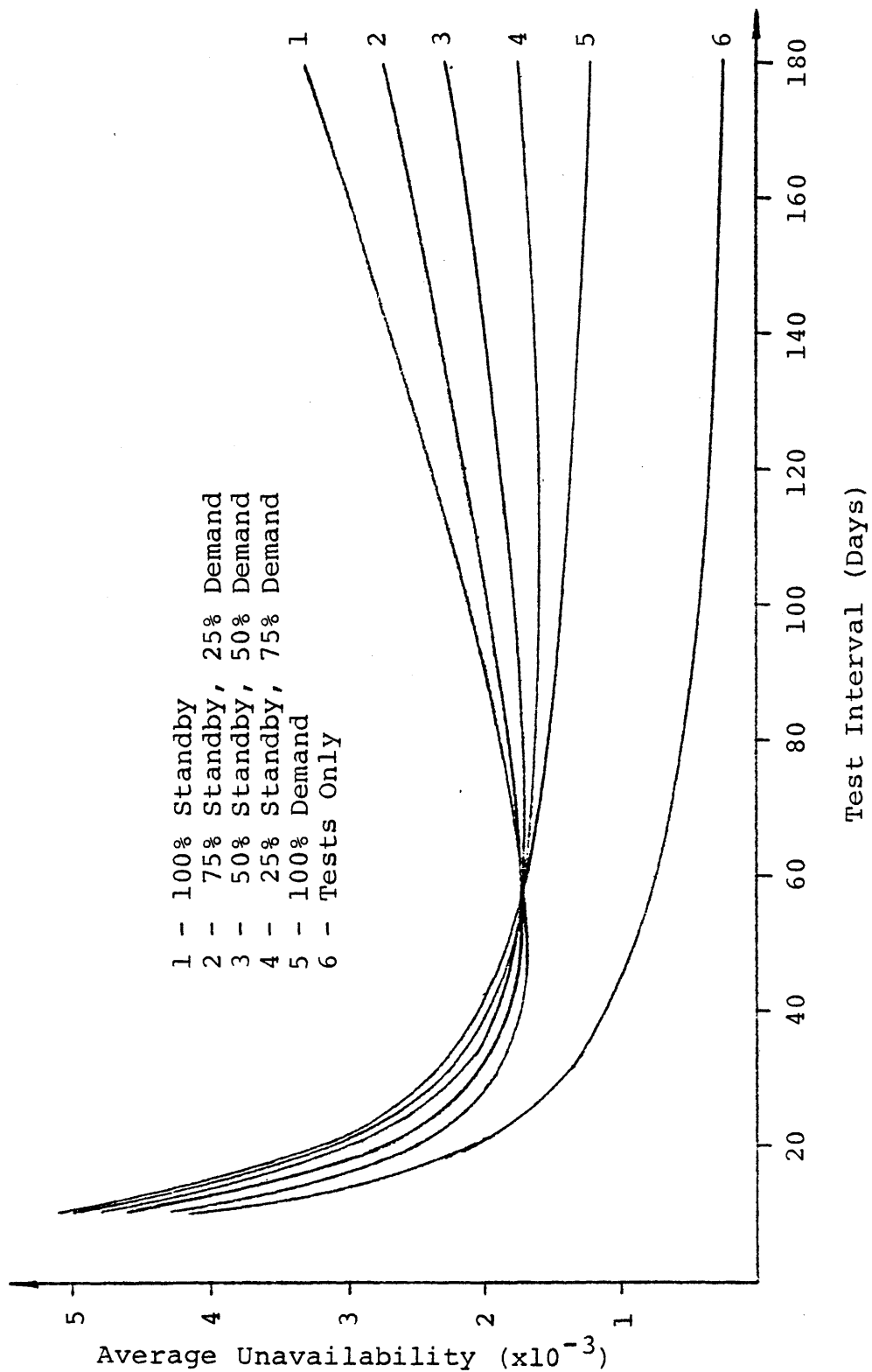


Figure 4.1. Sensitivity of Component Unavailability to Demand Verses Standby Failure Mechanisms, Constant Standby Failure Rate, Downtime per Test = 1 Hour.

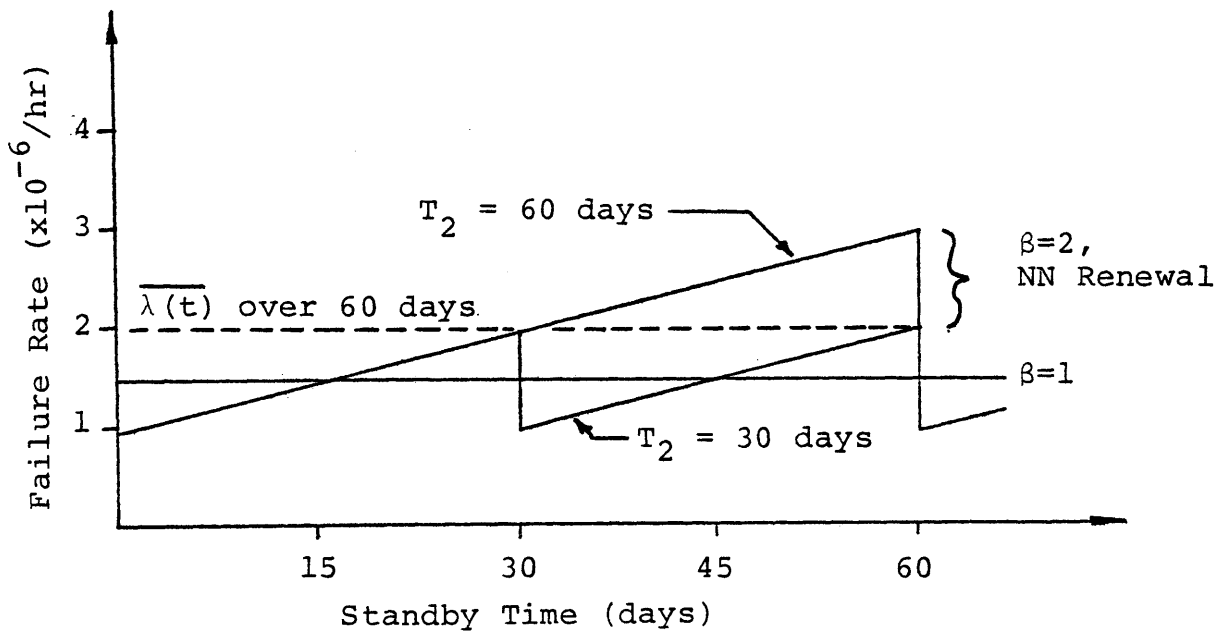
Where: $q_{\lambda}(T_2)$ = Instantaneous unavailability due to undetected standby failures at the time of the periodic test.

The contribution of each failure mechanism to the observed failure frequency is allowed to vary by 25% increments from all demand failures to all standby failures. The parameters for the curves are given in Table 4.1. The curves are generated by making repeated calculations for a given set of failure parameters while varying the test interval. The contribution due to testing alone is also shown. It increases as the test interval decreases reflecting the fact that the component is made unavailable for testing more often.

Figure 4.1 clearly shows the futility of periodic testing if only demand failure mechanisms are present and no renewal or maintenance benefit is obtained from the test. The 100% q_d curve is simply the sum of $q_d=0.001$ and the unavailability due to testing. For this case the best policy is to not test at all.

Beta = 1 Standby Failure Rate

Figure 4.1 was calculated assuming that the standby failure rate is constant in time during each calculation. When only standby failures are present, λ is at its largest value, and the unavailability curve has its largest dependence on test interval. Of course, as q_d increases, λ decreases, so the test interval at which the minimum una-



Effect of Renewal on the Failure Rates for Curves 1 in Figures 4.1 and 4.2

| Curve | q_d | $q_\lambda(T_2)$ | λ_0 | t_0 |
|--------------------------|---------|------------------|------------------------|---------|
| Figure 4.1 ($\beta=1$) | | | | |
| 1 | 0 | 0.001 | 1.39×10^{-6} | 0 |
| 2 | 0.00025 | 0.00075 | 1.04×10^{-6} | 0 |
| 3 | 0.00050 | 0.00050 | 6.95×10^{-7} | 0 |
| 4 | 0.00075 | 0.00025 | 3.47×10^{-7} | 0 |
| 5 | 0.00100 | 0 | 0 | - |
| Figure 4.2 ($\beta=2$) | | | | |
| 1 | 0 | 0.001 | 6.43×10^{-10} | 30 days |
| 2 | 0.00025 | 0.00075 | 4.82×10^{-10} | 30 days |
| 3 | 0.00050 | 0.00050 | 3.22×10^{-10} | 30 days |
| 4 | 0.00075 | 0.00025 | 1.61×10^{-10} | 30 days |
| 5 | 0.00100 | 0 | 0 | - |

Table 4.1. Component Failure Parameters for Figures 4.1, 4.2 and 4.3.

vailability is obtained becomes larger. It is also interesting to note that when the two types of failure mechanisms are evenly divided the curve at the minimum becomes quite flat while going through the minimum, indicating an insensitivity of the unavailability to test interval. There appears to be considerable latitude available to the systems analyst in choosing a test interval if demand failures form an appreciable percentage of the average unavailability and the standby failure rate is constant.

It is important to recognize that with a constant standby failure rate an increasing test interval does not produce more failures. The increase in the unavailability occurs because the failures that do occur remain undetected for a longer period of time. Provided that the approximation that $q_{\lambda} = \lambda T_2$ is valid, the same number of failures will be observed if the component is tested at two week as opposed to four week intervals. The probability of having a failure at two weeks is half that of four weeks, but there are twice as many two week tests.

Beta = 2 Standby Failure Rate, NN Renewal

Figure 4.2 shows the results of the same calculation as Figure 4.1 when a component has an increasing hazard rate and is renewed at every test and repair.² For comparison with Figure 4.1, the observed failure frequency for a test

² See Section 3.2.4 for a discussion of renewal types.

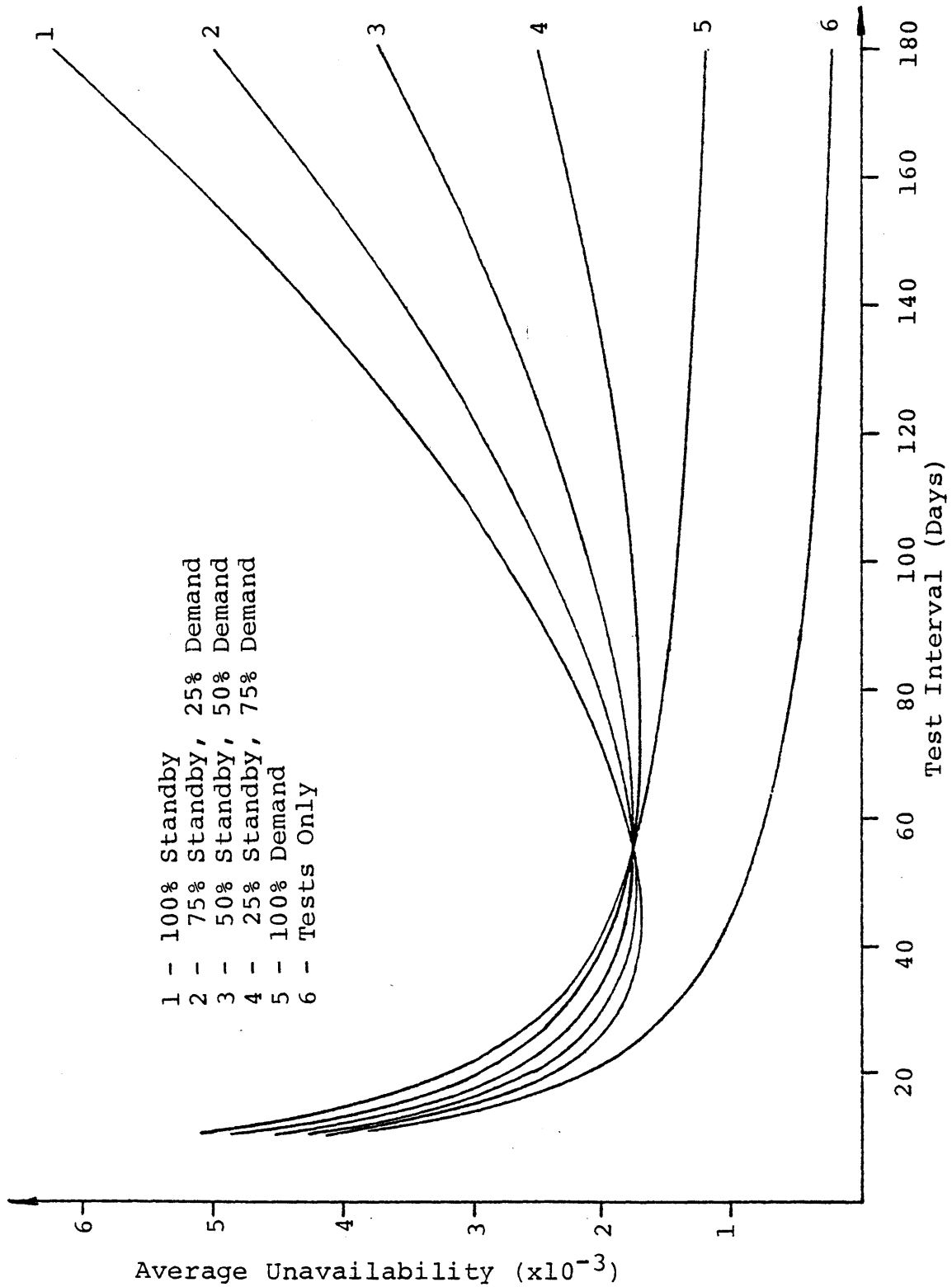


Figure 4.2. Sensitivity of Component Unavailability to Demand Verses Standby Failure Mechanisms, Beta = 2 Standby Failure Rate, Downtime per Test = 1 Hour.

interval of 30 days is again 0.001, with the division between observed failure mechanisms varying between all standby and all demand in 25% increments. The hazard rate has been modeled to increase linearly, doubling its value in the first 30 days of standby. Table 4.1 gives the component failure parameters used for this calculation.

When the hazard rate is increasing, the proper selection of a test interval becomes more important for at least two reasons:

- 1) As the standby period becomes longer the average unavailability increases at a faster rate than it did with the constant failure rate. This reflects the increasing susceptibility to failure which the increasing hazard rate models.
- 2) When the hazard rate increases with time during the standby period, more actual failures will result than if the component were tested more often. This can be easily understood by referring to the graphs of the hazard functions in Table 4.1. In order to obtain a failure probability of 0.001 at the 30 day test interval, the linear hazard rate is initially smaller than the constant rate. It crosses the constant rate at about 15 days, so that its average over the 30 day period is equal to the constant hazard rate. As the standby period becomes longer than 30 days, the linear hazard rate continues to increase. A component following this rate which has survived for 45 days without having been

tested at the 30 day point has a much higher probability of failing than one which was tested at 30 days, found working and returned to service. Again, the implication is that the test has not only verified that the component is working, but it has also reduced its probability of failure relative to what it would have been had there been no test. The average value of the hazard rate over a 60 day test period would be approximately the hazard rate's value at 30 days, whereas a 30 test interval would produce a hazard rate approximately equal to its value at 15 days.

4.1.2 EFFECT OF INCREASING DEMAND FAILURE RATE

Figure 4.3 shows the effect of increasing the demand failure rate in a component which has a specific standby failure rate. The standby failure rate produces an average unavailability of 0.001 when the test interval is 30 days. (The failure rate is double that of Figures 4.1 and 4.2.) These curves are plotted on semilog scale so that the percentage change in the component's unavailability with test interval can be seen more easily. It can be seen that as demand failures become more dominant the average unavailability of the component becomes less sensitive to the test interval. However, because the value of λ has remained constant, the optimum test interval for the component remains approximately the same.

4.1.3 EFFECT OF UNDETECTABLE FAILURES

In terms of their contribution to a component's una-

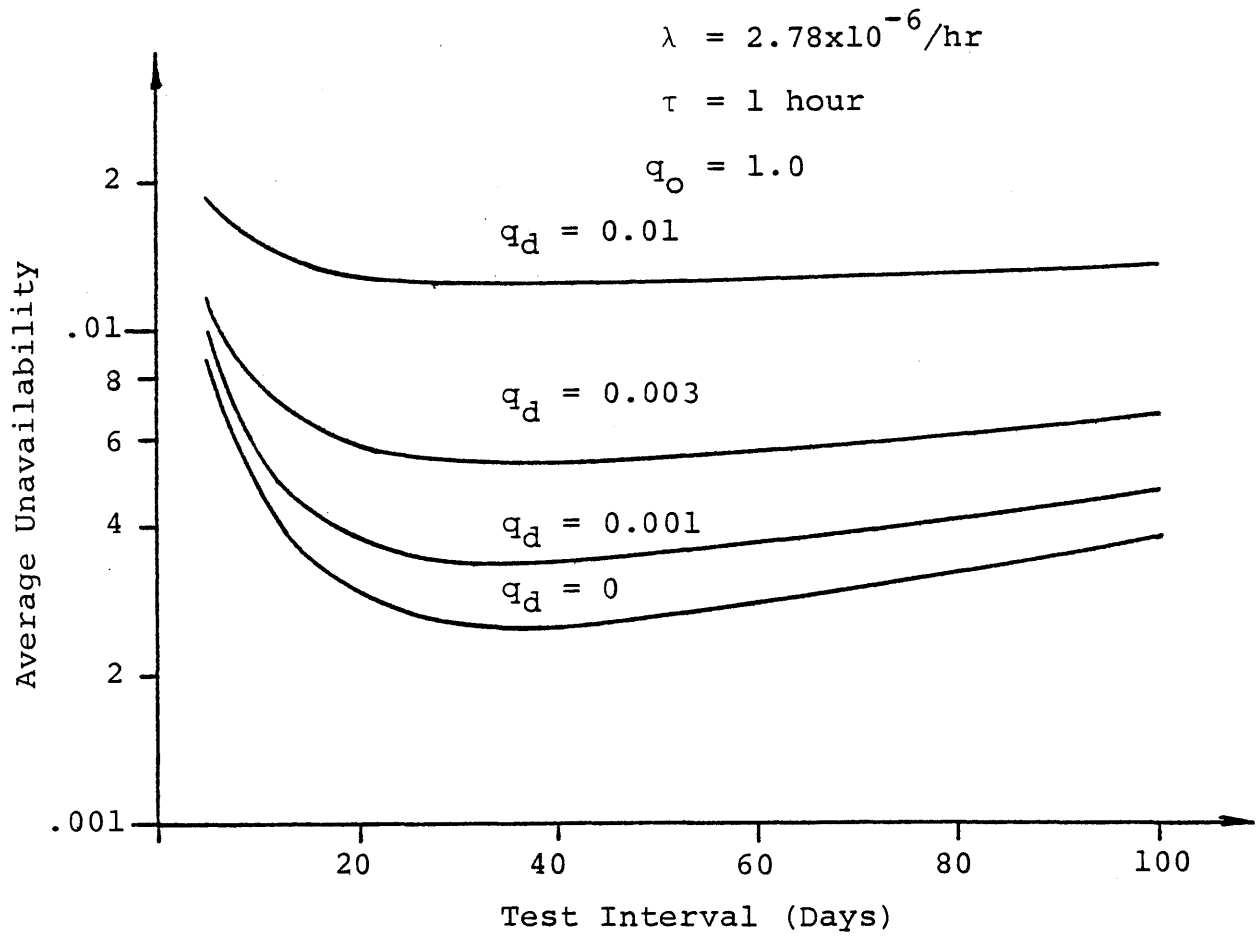


Figure 4.3. Effect of Periodic Testing on a Component With a Constant Standby Failure Rate and Various Magnitudes of Demand Failure Rate.

vailability during a test interval, undetectable failures are very much like demand failures. Their contribution to the unavailability is independent of the time since the last periodic test, since the test is incapable of revealing their existence. One can also give complementary physical interpretations to the two. Demand failures could be interpreted as the inherent inability of a component to cope with certain extreme conditions of a true demand. The value of q_d would then contain a contribution equal to the probability that the extreme conditions will exist at the time of the demand. (Given the conditions exist, the component's failure probability is one.) In the same context, one contributor to undetectable failures could result from a degradation of the component during its standby period which decreases its ability to respond to the extreme conditions of a true demand. In other words, undetectable failures can represent the time dependence of true demand failures.

For both the New-New and Old-Old renewal options of FRANTIC II-MIT, the unavailability due to undetectable failures is initially zero and increases monotonically throughout the calculation. For most practical applications the unavailability contribution will remain relatively constant throughout the interval, since undetectable standby failures should be small compared to those which the test can address. If, for example, its failure rate is constant the unavailability during any one test interval

changes by $\lambda_{\mu} T_2$ during any one interval. If demand failures are also possible, this increase will be a very small fraction of the test independent contribution to unavailability. Because of both its mathematical similarity to demand failures with in a test interval and its physical interpretation as a time dependent demand failure, there is no need to discuss the sensitivity of component unavailability to it any further.

4.1.4 FRANTIC CODE ASSUMPTION REGARDING DEMAND FAILURES

For periodically tested components in both FRANTIC II and FRANTIC II-MIT, it is assumed that all failure mechanisms modeled by q_d will produce demand failures at periodic tests that require repair. In other words, during the repair period there are two contributions due to q_d : 1) the component is under repair with a probability q_d due to demand failures at the previous test, and 2) the component which was good at the last test and is now on standby fails upon demand with probability q_d . Since q_d can model failures caused by conditions of the true demand, this assumption is not always correct. However, the error is conservative and small. For example, if q_d consists of entirely accident related failure probabilities, the actual average unavailability would be q_d , while the computed value would be $1 + (T_R/T_2)$. The additional factor is probably less than than 10% for most repair times and test intervals of interest.

If the analyst does not want demand failures to make a repair contribution to the unavailability, he can always establish a composite component using two failure events combined using an OR gate. In one event only q_d is input, and there will be no repair period. The second will have only the periodically tested failure parameters. The resultant composite component will not have the contribution.

4.2 SUBROUTINE OPTEST

4.2.1 BACKGROUND

The determination of an optimum component test interval for a component using the original FRANTIC II is very inefficient. A series of calculations must be made inputting different values of the periodic test interval for each calculation. A curve of average system unavailability verses periodic test interval can then be plotted from which the minimum value can be obtained. This procedure is both time consuming and costly. Subroutine OPTEST provides a quick way to calculate the optimum test interval, minimum unavailability, and variation of the average unavailability with test interval of any system that can be represented by a single set of failure parameters.

4.2.2 THEORY

The derivation of the equations used in OPTEST could procede directly from Equations (3.1) through (3.3). Howev-

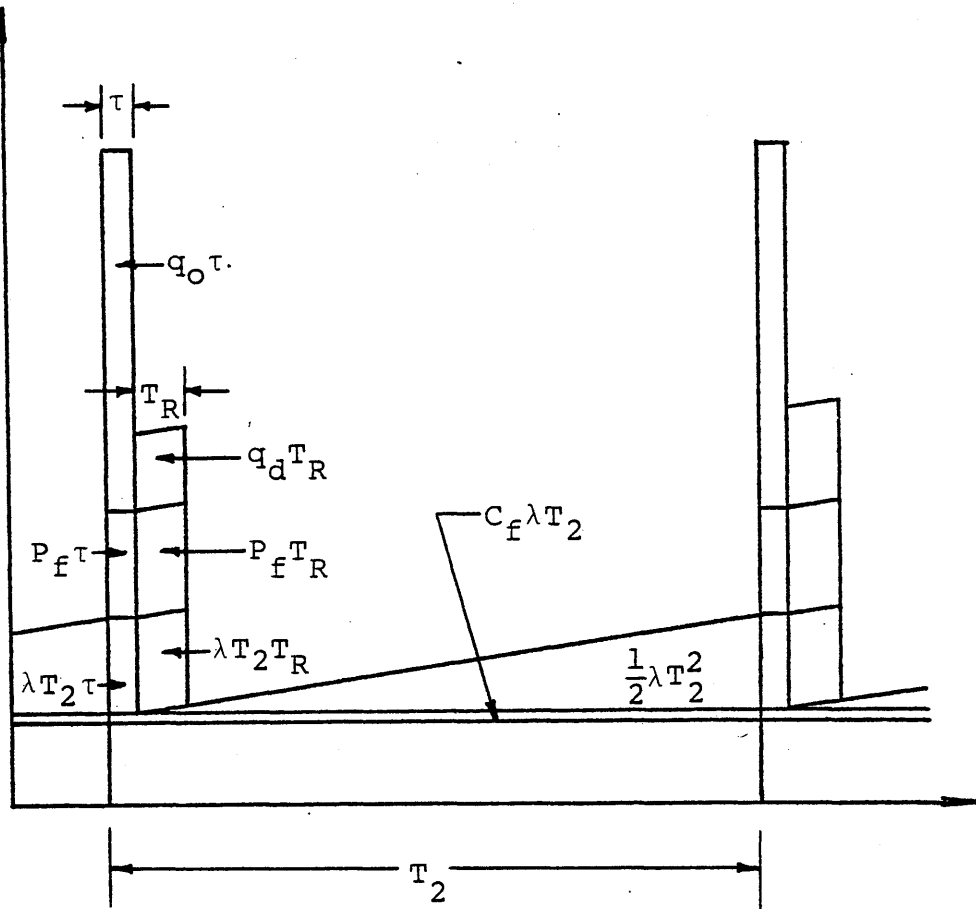


Figure 4.4. Graphical Representation of the Time Integrated Unavailability of a Periodically Tested Component.

er, it can be most easily understood by referring to the graphical representation of time dependent component unavailability presented in Figure 4.4. This figure is the same as Figure 2.2. However, here the areas under the curve are emphasized, because they represent the contributions of the various possible failure modes to the component's time integrated unavailability,

$$\bar{q}_c = \frac{1}{T} \int_0^T q_c(t) dt \quad (4.2)$$

In terms of Figure 4.4, this equation can be written as

$$\begin{aligned} \bar{q}_c = \frac{1}{T} [& \frac{1}{2} \lambda T^2 + C_f \lambda T^2 + q_d T + \lambda T \tau + q_o \tau \\ & + P_f \tau + \lambda T T_R + P_f T_R + q_d T_R] \end{aligned} \quad (4.3)$$

In writing this equation, three simplifying assumptions are made:

1. $T_s \gg \tau$, so that $T_s = T_2 - \tau \approx T_2$. This assumption produces a slightly larger unavailability due to undetected standby failures than would be expected.

2. The probability of undetected/unrepaired standby failures is assumed to remain less than 0.01, so that $q_\lambda(T_2) \approx \lambda T_2$.

3. The accumulation of the probability of undetected standby failures is assumed to begin at the end of the test

period independent of the probability that the component will require unscheduled repair. This assumption also produces a slightly larger unavailability than would actually be experienced.

3. The carryover factor, C_f , is assumed to model failures that will go undetected for only one test. That is, component failures which go undetected for one periodic test will be detected at the next test. It is recognized that this is an unconservative assumption. However, the test should be separated in time sufficiently for the human error probabilities to be independent, and a review of test procedures which reveals that there is a potential for C_f to be greater than about 0.01 should result in a revised procedure rather than an input to a fault tree. Therefore, the probability of successive nondetection of failures should be very small compared to the probability of just one nondetection. It should also be noted that input parameters for undetectable failure modes might be the proper vehicle to model modes which could suffer repeated nondetection.

As shown in Equations (3.1) through (3.3) of Chapter 3, the contributions to unavailability during standby, testing, or unscheduled repair can not be simply added, since they are mutually exclusive. Applying the proper factors to the failure modes which could occur together in the standby, test and repair periods respectively and collecting terms produces the following equation:

$$\bar{q}_c = AT + B + C/T \quad (4.4)$$

Where:

$$A = (1-q_d)(0.5+C_f)\lambda \quad (4.4a)$$

(Note: In this derivation, it is assumed that undetected and carryover failures will occur with a low enough probability to use the rare event approximation when writing (4.4a). A crossterm would add considerable complexity to the mathematical presentation.)

$$B = q_d + (1-q_d)[\tau(1-q_o)(1-P_f)\lambda + T_R(1-q_d)(1-P_f)\lambda] \quad (4.4b)$$

$$C = (1-q_d)[\tau(q_o + \{1-q_o\}P_f) + T_R(q_d + \{1-q_d\}P_f)] \quad (4.4c)$$

The interaction of the various failure modes with the periodic test interval may now be easily interpreted.

- Coefficient A contains those failure modes whose contribution to system unavailability can be reduced by reducing the periodic test interval. These are the contributions due to undetected standby failures, both those which occur during the current standby interval and those which escaped detection during the previous periodic test.

- Coefficient B contains those failure modes whose contributions to average system unavailability are independent of the test interval. These include the demand failure mode and the downtime produced when standby failures are detected and repaired during period tests. (If the tests are further apart in time, each one is more likely to produce a failure, but since they are further apart, the more likely event occurs less often. These two factors cancel.)
- Coefficient C contains those failure modes whose contributions are decreased by increasing the test interval. These include the contributions of imperfect testing, namely unavailabilities due to unscheduled repair of test caused failures, inability to override the test mode in the event of a demand to perform the safety function, and unscheduled repair of demand failures resulting from the cycling to test mode.

The optimum test interval can be found by taking the first derivative of equation (4.4) and setting it equal to zero, which produces:

$$T_{Op} = \sqrt{\frac{C}{A}} = \sqrt{\frac{\tau[q_0 + (1-q_0)P_f] + T_R[q_d + (1-q_d)P_f]}{0.5(1+C_f)\lambda}} \quad (4.5)$$

As expected, those contributions resulting from imperfect testing increase the optimum test interval, while those which result in undetected standby failures decrease it. Note that Equation (4.5) reduces directly to Equations (2.11) and (2.13) if $q_o = 1$ and all parameters except τ and λ are assumed to be zero.

An expression for the minimum unavailability can be obtained by using Equation (4.5) in Equation (4.4). This results in:

$$\bar{q}_c(\min) = AT_{op} + B + \frac{T_{op}}{C} \quad (4.6)$$

Additional flexibility can be obtained by rearranging Equation (4.3) into quadratic form:

$$AT^2 + (B - \bar{q}_c)T + C = 0 \quad (4.7)$$

The average unavailability can now be any value. For convenience it is expressed as a factor increase f over the minimum average unavailability. Since the minimum unavailability occurs at the optimum test interval:

$$q_c = f\bar{q}_c(\min) = f\left[A\sqrt{\frac{C}{A}} + B + C\sqrt{\frac{A}{C}}\right] \quad (4.8)$$

Where f is defined by:

$$f = \frac{\bar{q}_c(T_2, \text{failure parameters})}{\bar{q}_c(\text{min})(T_{\text{op}}, \text{failure parameters})} \quad (4.9)$$

This equation reduces to:

$$\bar{q}_c = f[B + 2\sqrt{AC}] \quad (4.10)$$

Using equation (4.10) as the value of the average unavailability in equation (4.7) and applying the quadratic formula, one obtains:

$$T_2 = \frac{2f\sqrt{AC} + (f-1)B \pm \sqrt{[(1-f)B - 2f\sqrt{AC}]^2 - 4AC}}{2A} \quad (4.11)$$

The quantity T_2 is now the periodic test interval at which the minimum average system unavailability is increased by a factor of f over what it would be at the optimum periodic test interval. With this equation one can determine the range of period test intervals which will produce no more than a factor of f increase in the component's unavailability over that obtained at the optimum test interval. For most practical applications one would be interested in lengthening the test interval, as this will decrease the potential for system wearout and will decrease manpower requirements.

It may be of interest to determine how far one is from the optimum periodic test policy for a given set of failure parameters and test interval. This can be done by calculating f using Equation (4.9). First, the optimum test interval for a set of failure parameters is calculated using Equation (4.5), and the minimum average unavailability obtained by testing at that interval is determined using Equation (4.6). The component's unavailability is then recalculated using Equation (4.4) for the specific test interval that was input. This quantity is then divided by the original minimum unavailability to obtain the factor increase created by not testing at the optimum interval.

4.2.3 IMPLEMENTATION IN FRANTIC II-MIT

The theory presented in the section above is implemented in FRANTIC II as subroutine OPTTEST. Component failure parameters can be input using the COMP keyword, and OPTTEST is called using the TEST keyword. The user must provide the index number of the component he wishes to analyze, the calculation option for OPTTEST, and (optionally) the factor used to find the range of test intervals producing less than this factor increase in unavailability. (In the absence of a factor the default value of 1.1 is used.)

Two calculation options are provided.

- 1) Option 1 calculates the optimum test interval, the unavailability at this interval (q_{\min}), and the range of test

interval producing less than a factor f increase in the unavailability.

2) Option 2 calculates the optimum test interval, the unavailability at this interval, the unavailability at the user input test interval, T_2 , in the component input data, and the factor increase, f , produced by testing at the user input interval.

For a detailed description of the calculation sequence, the reader is referred to a listing of SUBROUTINE OPTTEST, which is fully documented with comments. The format for the OPTTEST keyword input is included in Appendix A.

4.2.4 COMPARISON WITH FRANTIC II-MIT RUN OPTION

Figure 4.5 is a comparison of OPTTEST calculations with those of a standard FRANTIC II-MIT calculation. It can be seen from Figure 4.5 that excellent agreement is obtained between the two methods of calculating unavailability for a single component. These results are also in excellent agreement with Figures 2.4 [Ja68] and 2.5 [Lo81].

Table 4.2 shows the input and output of the OPTTEST calculation. With OPTTEST the optimum test interval is given directly, as well as information about the sensitivity of the unavailability to changes in the test interval. The optimum test interval of 18.6 days and minimum unavailability of $4.47E-3$ are in excellent agreement with the values of 447 hours and 0.004472 from Table 3 of Lofgren's work.

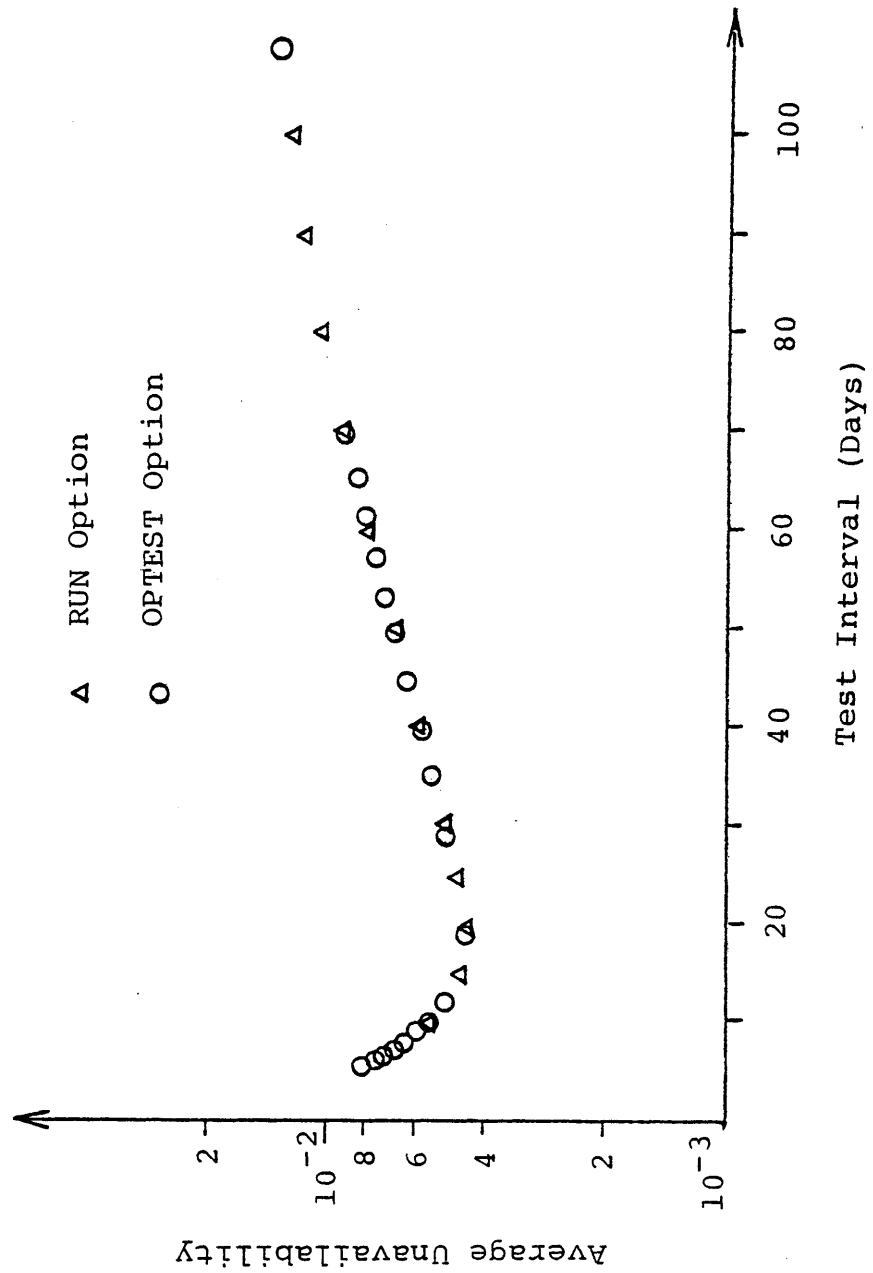


Figure 4.5. Comparison of OPTEST and RUN Calculations of the Average Unavailability of a Periodically Tested Component Verses Test Interval.

FILE: TABLE42 DAT A

TITLE
 FIGURE 4.5 COMPARISON OF OPTEST AND RUN
 POPT

COMP
 NEW
 1 1COMP 10.0 30.0 29. 1.0 1.0
 1 2COMP 1.0 1.0
 -1
 TEST
 1 1 1.1
 1 1 1.2
 1 1 1.3
 1 1 1.4
 1 1 1.5
 1 1 1.6
 1 1 1.7
 1 1 1.8
 1 1 1.9
 1 1 2.0
 1 1 3.0
 1 1 4.0
 1 1 5.0
 2 1

FILE: TABLE42 OPTEST A

| OPTION 1 | | | | | | | | | | | |
|----------|----------|--------|---------|------|-------|--------|----------|----------|------|-------|-------|
| J | LAMBDA | QRESID | TAU | TREP | QOVR | PTCF | F | QMIN | T2OP | T2LOW | T2HI |
| | | | (HOURS) | | | (DAYS) | | | | | |
| 1 | 1.00E-05 | 0.0 | 1.0 | 0.0 | 1.000 | 0.0 | 1.10E+00 | 4.47E-03 | 18.6 | 12.0 | 29.0 |
| 1 | 1.00E-05 | 0.0 | 1.0 | 0.0 | 1.000 | 0.0 | 1.20E+00 | 4.47E-03 | 18.6 | 10.0 | 34.7 |
| 1 | 1.00E-05 | 0.0 | 1.0 | 0.0 | 1.000 | 0.0 | 1.30E+00 | 4.47E-03 | 18.6 | 8.7 | 39.7 |
| 1 | 1.00E-05 | 0.0 | 1.0 | 0.0 | 1.000 | 0.0 | 1.40E+00 | 4.47E-03 | 18.6 | 7.8 | 44.3 |
| 1 | 1.00E-05 | 0.0 | 1.0 | 0.0 | 1.000 | 0.0 | 1.50E+00 | 4.47E-03 | 18.6 | 7.1 | 48.8 |
| 1 | 1.00E-05 | 0.0 | 1.0 | 0.0 | 1.000 | 0.0 | 1.60E+00 | 4.47E-03 | 18.6 | 6.5 | 53.1 |
| 1 | 1.00E-05 | 0.0 | 1.0 | 0.0 | 1.000 | 0.0 | 1.70E+00 | 4.47E-03 | 18.6 | 6.1 | 57.3 |
| 1 | 1.00E-05 | 0.0 | 1.0 | 0.0 | 1.000 | 0.0 | 1.80E+00 | 4.47E-03 | 18.6 | 5.7 | 61.4 |
| 1 | 1.00E-05 | 0.0 | 1.0 | 0.0 | 1.000 | 0.0 | 1.90E+00 | 4.47E-03 | 18.6 | 5.3 | 65.5 |
| 1 | 1.00E-05 | 0.0 | 1.0 | 0.0 | 1.000 | 0.0 | 2.00E+00 | 4.47E-03 | 18.6 | 5.0 | 69.5 |
| 1 | 1.00E-05 | 0.0 | 1.0 | 0.0 | 1.000 | 0.0 | 3.00E+00 | 4.47E-03 | 18.6 | 3.2 | 108.6 |
| 1 | 1.00E-05 | 0.0 | 1.0 | 0.0 | 1.000 | 0.0 | 4.00E+00 | 4.47E-03 | 18.6 | 2.4 | 146.7 |
| 1 | 1.00E-05 | 0.0 | 1.0 | 0.0 | 1.000 | 0.0 | 5.00E+00 | 4.47E-03 | 18.6 | 1.9 | 184.5 |
| OPTION 2 | | | | | | | | | | | |
| J | LAMBDA | QRESID | TAU | TREP | QOVR | PTCF | Q(AT T2) | QMIN | T2OP | T2 | F |
| | | | (HOURS) | | | (DAYS) | | | | | |
| 1 | 1.00E-05 | 0.0 | 1.0 | 0.0 | 1.000 | 0.0 | 4.99E-03 | 4.47E-03 | 18.6 | 30.0 | 1.12 |

Table 4.2. Input and Output of OPTEST Calculations for Figure 4.5.

FILE: TABLE43 DAT A

TITLE
 FIGURE 4.5 COMPARISON OF OPTEST AND RUN
 POPT

```

COMP
NEW
  1 1COMP    10.0          5.0  4.    1.0    1.0
  1 2COMP     1.0                1.0
-1
TIME
100.          2.
RUN
  1 TOTL NONE          FIGURE 4.5 COMPARISON OF OPTEST AND RUN
-1
COMP
NEW
  1 1COMP    10.0          10.0  9.    1.0    1.0
  1 2COMP     1.0                1.0
-1
TIME
200.0        3.
RUN
  1 TOTL NONE
-1
COMP
NEW
  1 1COMP    10.0          15.0 14.    1.0    1.0
  1 2COMP     1.0                1.0
-1
TIME
300.          5.
RUN
  1 TOTL NONE
-1

```

(Note: Input continues for 170 more lines)

FILE: TABLE43 FILE A

| LAMBDA(1) | OFFSET(1) | BETA(1) | QRESID(1) | TEST2(1) | QSAVG |
|-----------|-----------|----------|-----------|----------|----------|
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 5.00E+00 | 8.91E-03 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 1.00E+01 | 5.35E-03 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 1.50E+01 | 4.55E-03 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 2.00E+01 | 4.46E-03 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 2.50E+01 | 4.64E-03 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 3.00E+01 | 4.96E-03 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 3.50E+01 | 5.36E-03 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 4.00E+01 | 5.80E-03 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 4.50E+01 | 6.28E-03 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 5.00E+01 | 6.79E-03 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 5.50E+01 | 7.31E-03 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 6.00E+01 | 7.84E-03 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 6.50E+01 | 8.38E-03 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 7.00E+01 | 8.93E-03 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 7.50E+01 | 9.48E-03 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 8.00E+01 | 1.00E-02 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 8.50E+01 | 1.06E-02 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 9.00E+01 | 1.12E-02 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 9.50E+01 | 1.17E-02 |
| 1.00E-05 | 0.0 | 1.00E+00 | 0.0 | 1.00E+02 | 1.23E-02 |

Table 4.3. Input and Output of RUN Calculations for Figure 4.5.

Table 4.3 shows the input and output of the RUN option calculation. Because of its size (205 lines), all the input for the RUN option is not given. As FRANTIC II-MIT does not contain a search routine for optimization, a series of calculations must be done to determine where the unavailability goes through a minimum. The RUN option is designed for use with complex systems, so its instantaneous unavailability and numerical integration algorithm is inefficient for single components. Its flexibility for addressing multicomponent system's far outweigh this minor inconvenience for one component systems, as will be demonstrated in Chapters 5 and 6. OPTEST fills the gap for constant failure rate single components.

4.3 BEHAVIOR OF CONSTANT FAILURE RATE COMPONENTS

The analytical form of the unavailability equations used in OPTEST lead to insights which can be quite useful in establishing a periodic testing program. The following sections give a few examples.

4.3.1 APPROXIMATE RELATIVE IMPORTANCE OF FAILURE MODES

Examination of Equation (4.4) shows that the relative importance of the various component failure modes is influenced by their duration. For the purposes of comparison, it would be useful to express the test and repair times in terms of the period test interval.

Say that test time is typically about one thousandth of the periodic test interval. This corresponds to a test time of slightly less than one hour when the period test is done monthly. Also suppose that the repair time is approximately one fiftieth of the test interval, corresponding to repair taking an average of 15 hours. Also assume $q_d < 0.01$, so that $(1-q_d) \approx 1$. Equation (4.4) then becomes:

$$\begin{aligned} \bar{q}_c = & (0.5+C_f)\lambda T + q_d + 10^{-3}T(1-q_o)(1-P_f)\lambda + \\ & 0.02T(1-P_f)\lambda + 10^{-3}q_o + 10^{-3}(1-q_o)P_f + \\ & 0.02q_d + 0.02P_f \end{aligned} \quad (4.12)$$

Collecting terms obtains:

$$\bar{q}_c = [0.521+C_f]\lambda T + 10^{-3}q_o + 0.021P_f + 1.02q_d \quad (4.13)$$

With the equation in this form, the following observations can be made:

1) The coefficient multiplying the standby failure contribution due to undetected failures awaiting testing to be revealed is 0.5. A contribution of 0.021 in the coefficient is due to detected standby failures during periodic testing and unscheduled repair. It can be seen that this contribution is small compared to the undetected contribution.

2) The coefficient of q_o reflects the relatively short period of time that the component is undergoing periodic

testing. Consequently the unavailability to override the test mode must be quite high for it to become comparable to the contributions of standby and demand failures.

3) The coefficient of P_f is larger than that of q_0 because test caused failures must be repaired. Consequently, the component will be unavailable for a longer period of time. However, the coefficient is still quite small, reflecting again the relatively short duration of the test and repair periods compared to the test interval.

4) The demand failure rate is multiplied by one, because a demand failure can occur whenever a demand is made to accomplish a test or safety function. In fact, it is slightly greater than one because demand failures which occur at the beginning of a periodic test must be repaired.

5) The crossterms reflect the fact that the different failure modes combine through an OR gate. The sum of the first four terms would be maximum unavailability obtained by a combination of all failure modes.

4.3.2 EFFECTIVE TEST DOWNTIME

The coefficient C of Equation (4.4) can be given a physical interpretation which will make it easier to use OPTTEST for engineering applications. Recall that this coefficient is given by Equation (4.4c). If the terms which account for the fact that they are mutually exclusive are removed, the contributions to this term become more clear:

$$C = \tau(q_o + P_f) + T_R(q_d + P_f) \quad (4.14)$$

Each term is just a probability that the component is failed times the length of time it will be down during test or repair due to this cause. All four causes are created by the test.³ It is interesting to note that the time integral of all these contributions carry the same weight in determining the optimum test interval. For the purpose of establishing a testing policy for a one component system it will be useful to consider them as one group of contributors to the Effective Downtime per Test.

The Effective Downtime per Test is very useful for determining the unavailability caused by testing. equally.⁴

For example, if testing of a component takes 0.5 hours during which it is entirely unavailable and generates test caused failures at a rate of 0.05 per test which require on

³ For periodically tested components, FRANTIC II-MIT assumes that all demand failures can occur at the test demand and require repairs whose duration is the same as for standby failures (see Section 4.1.4). With the exception of a test caused change in the failure rate (see Section 3.5.3), no benefit is obtained by testing for demand failures, because the probability that they will occur again after the test is not changed.

⁴ For a single component system these contributions are weighted. For multiple component systems the unavailability of other components affect the importance of a component's unavailability to the system. For example there is no penalty for taking a component offline if a redundant component is known to be working.

the average of 20 hours to repair, then, on the average, the EDT per test is:

$$\text{EDT} = (1.0)0.5 + (0.05)20 = 0.6 \text{ hours/test} \quad (4.15)$$

(Note that the maximum unavailability during the test is 1.0.) The EDT can then be used directly in tables and graphs such as those given by Lofgren [Lo80] and the unavailability contours which are introduced in the next section.

4.3.3 UNAVAILABILITY CONTOURS

Figure 4.6 is a mapping of unavailability verses test interval for a range of failure rates. It was derived by using option 1 of OPTTEST. Calculations such as those shown in Table 6.2 were made for various values of λ . All the q_{\min} data points, $1.1q_{\min}$ data points, etc., were connected to form the contour lines shown.

Figure 4.6 can be read much like a contour map, yielding a "valley of unavailability" about the optimum test interval. Note that the valley can extend in two directions. For a given value of λ , say $1.0\text{E-}6/\text{hr}$, a range of test intervals come within 10% of the minimum unavailability would lie between 38 to 92 days when the EDT is 1.0 hr/test. Conversely, if one were to decide to test a component at thirty day intervals, he would be within 10% of the optimum test interval for values of λ in the range of 1.6 to $9\text{E-}6/\text{hr}$.

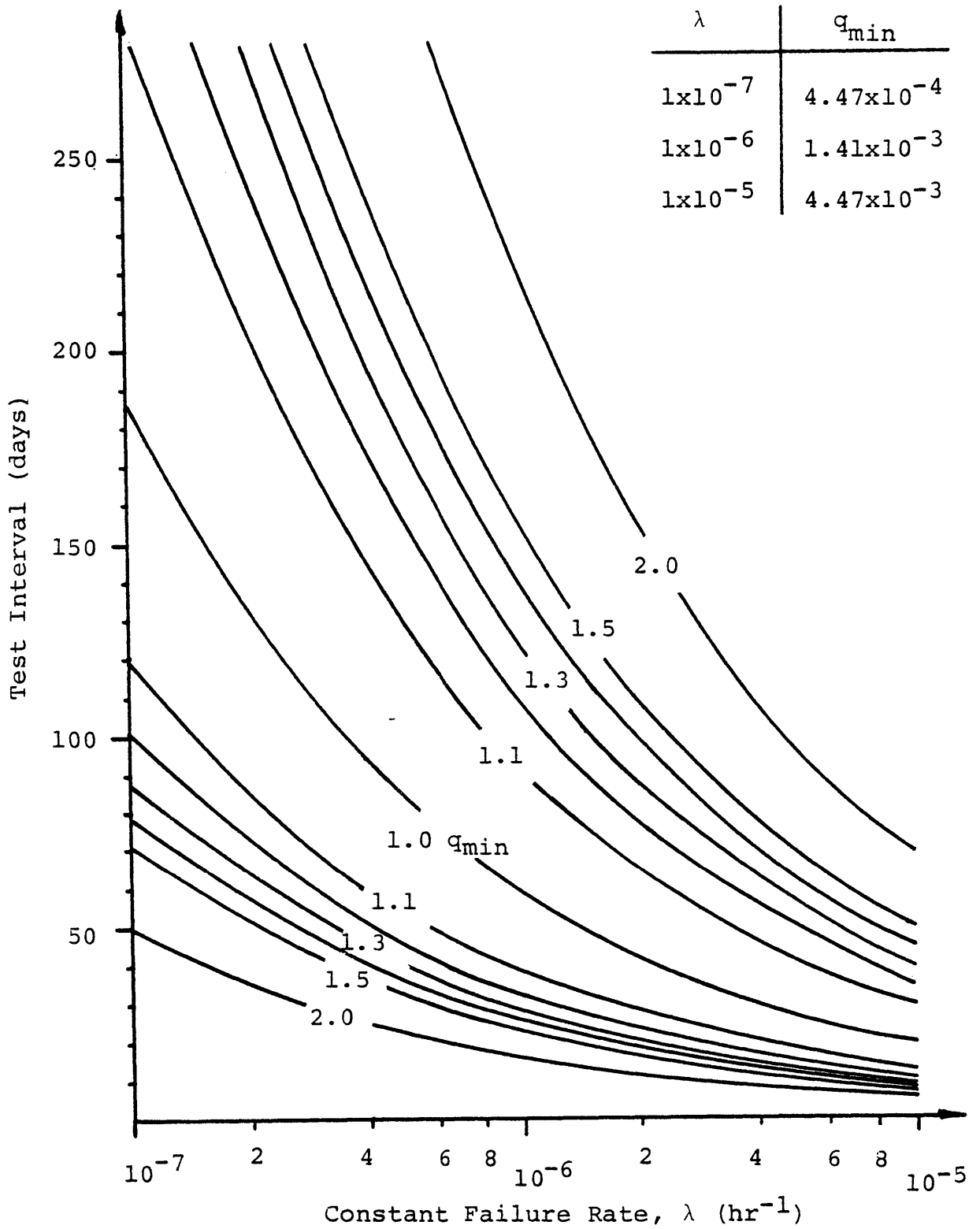


Figure 4.6. Average Unavailability Contours for a Periodically Tested Component Having an Effective Downtime Per Test of One Hour.

It is interesting to note that as the value of the standby failure rate increases, the width of the "valley" decreases. This should be expected, since at the higher failure rates more failures occur per unit time, so the testing policy ought to become more critical. For failure rates below say $1E-6/hr$, the $1.1q_{min}$ contours embrace a choice of test intervals ranging from 38 to 92 days, a total of 54 days, whereas when $\lambda = 1E-5/hr$ the range has declined to from 12 to 29 days, or a 17 day interval.

4.4 BEHAVIOR OF TIME DEPENDENT HAZARD RATE COMPONENTS

With the introduction of the offset time, the flexibility available in FRANTIC II-MIT to model hazard rates which vary with time makes the presentation of sensitivity calculations to illustrate the general characteristics of the resultant unavailability impractical. Instead, a few comments will be made about how a time dependent failure rate might effect a calculaton differently than a constant failure rate.

4.4.1 IMPORTANCE OF HAZARD RATE

Figure 4.7 illustrates how the time dependence of the hazard rate can influence the rate of change of the instantaneous unavailability. The calculation is made for an increasing, constant, and decreasing hazard rate. The hazard rates are normalized so that they give approximately the same instantaneous unavailability after a 30 days standby

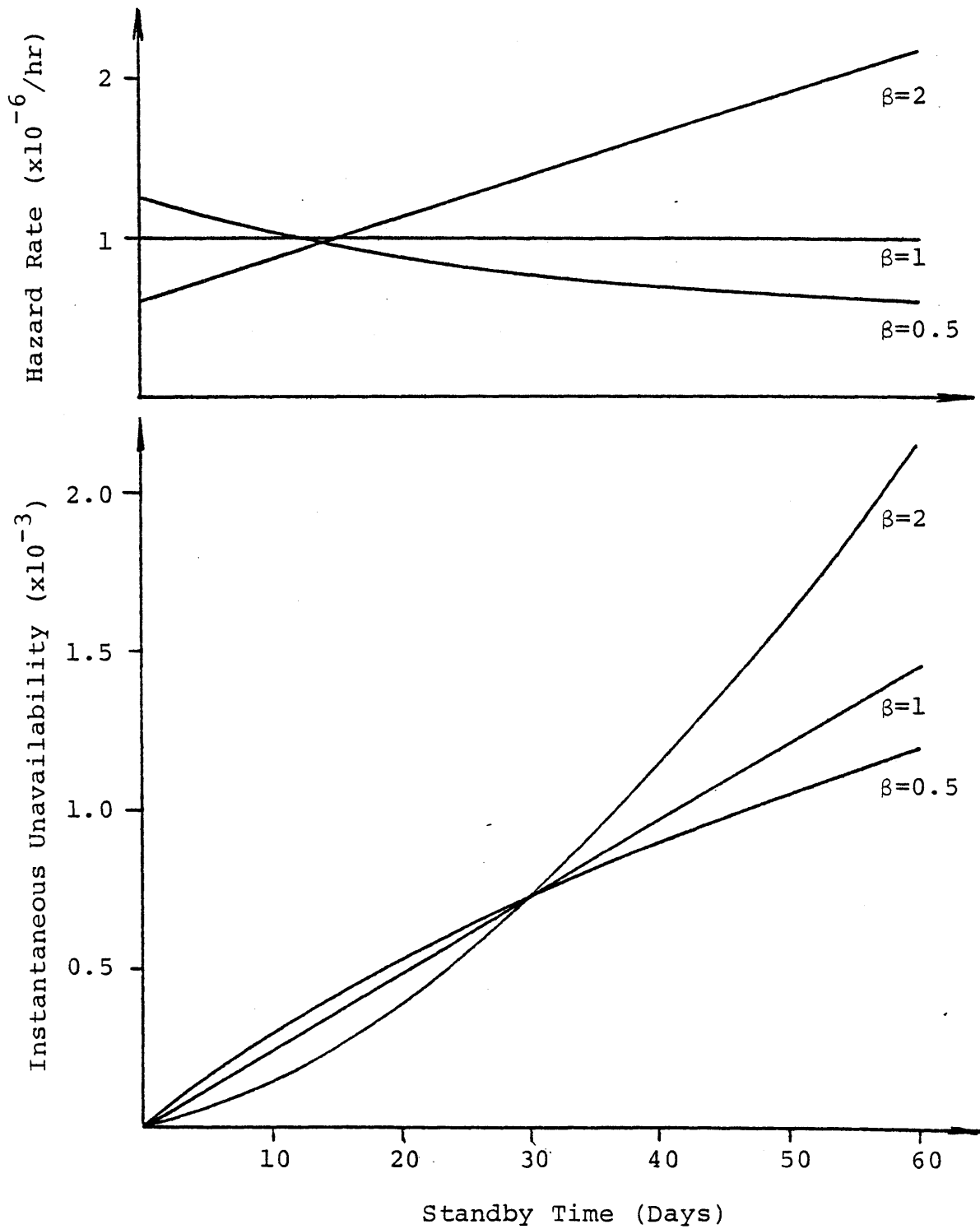


Figure 4.7. Instantaneous Unavailability Resulting From Hazard Rates Having Three Different Time Dependencies.

period. The calculation is then carried out to 60 days to see the effect of the individual time dependences. The top half of the figure shows the hazard rates and the bottom half shows the resultant instantaneous unavailabilites.

For values of the instantaneous unavailability less than about 0.01 (so that there is a high probability of surviving to come under hazard over the entire period) there is a direct correspondence between the area under the hazard curve and the magnitude of the instantaneous unavailability. Therefore, an increasing hazard rate will increase the rate at which the instantaneous unavailability rises, while the decreasing rate causes it to begin to level off.

4.4.2 USES OF RENEWAL OPTIONS

New-New Renewal

With New-New renewal both test and repair reset the hazard rate. For these conditions, Figure 4.7 shows that the proper selection of a test interval becomes more important as the value of β increases. A typical example of how the New-New renewal option is used with an increasing failure rate was given in Section 4.1.1, so the effect of the use of this option will not be discussed other than to comment about a decreasing failure rate. It is hard to visualize a physical situation in which a component would be modeled with a decreasing failure rate and a NN renewal option. The only plausible situation that this author can envision is

that somehow a test creates a condition from which a component must recover during standby. When it initially comes off test, it is quite susceptible to the random shocks of standby. But as it sits idle it can somehow take the shocks better, or perhaps the rate of shocks die off following the test. Data which implies this type of condition exists should be considered very carefully.

Old-New Renewal

Recall that the Old-New renewal option resets the hazard function back to the offset time if a component is failed at a given test. In the code this calculation becomes very involved, because every test has a probability of producing failure, so the instantaneous unavailability depends on a composite of hazard rates extending back to the beginning of the calculation. A decreasing hazard rate would model replacements where there are possibilities of substandard components. The longer a component survives, the less chance of it being substandard. The effect will be a gradually declining unavailability.

If the hazard rate increases, in theory a steady state will be reached. Those components which have survived for a long time will have very large hazard rates and will almost surely fail. They will then be replaced with components whose hazard rate starts at the offset time. Eventually there will be a balance. In practice, an increasing failure rate with replacement upon failure is an excellent candidate

for the application of renewal theory. When the components are judged to be very likely to fail, they should be replaced. FRANTIC II-MIT does not currently have a model to reflect replacement policies such as this.

Old-Old Renewal

There are two applications for which the Old-Old renewal option can be used, modeling long term wear-out or burn-in and investigating maintenance policies.

Long Term Wear-out or Burn-in

Old-Old renewal means that neither test nor repair resets the hazard rate. It implies a hazard function that gradually changes over the course of many test intervals. The hazard rate during any one test interval remains relatively constant, and the unavailability during that interval behaves much like it would for a constant failure rate equal to the value of the hazard rate at the middle of the interval. Table 4.4 illustrates this. It gives the results of calculations comparing the average annual unavailability due to a $\beta=3$ hazard rate with a constant hazard rate. The $\beta=3$ hazard rate is allowed to vary over a 20 year time period, and calculations are made using the offset time to establish the year in the which the calculation is being made. For comparison a constant ($\beta=1$) hazard rate is calculated at the midpoint of each year and run separately. It can be seen from Table 4.4 that very close agreement is obtained between

| BETA | LAMBDA | OFFSET (YEARS) | TEST INT (DAYS) | OSAVG |
|----------|----------|-------------------|--------------------|----------|
| 3.00E+00 | 1.00E-15 | 0.0 | 3.00E+01 | 2.67E-05 |
| 3.00E+00 | 1.00E-15 | 1.00E+00 | 3.00E+01 | 1.90E-04 |
| 3.00E+00 | 1.00E-15 | 2.00E+00 | 3.00E+01 | 5.16E-04 |
| 3.00E+00 | 1.00E-15 | 3.00E+00 | 3.00E+01 | 1.01E-03 |
| 3.00E+00 | 1.00E-15 | 4.00E+00 | 3.00E+01 | 1.66E-03 |
| 3.00E+00 | 1.00E-15 | 5.00E+00 | 3.00E+01 | 2.47E-03 |
| 3.00E+00 | 1.00E-15 | 6.00E+00 | 3.00E+01 | 3.45E-03 |
| 3.00E+00 | 1.00E-15 | 7.00E+00 | 3.00E+01 | 4.59E-03 |
| 3.00E+00 | 1.00E-15 | 8.00E+00 | 3.00E+01 | 5.88E-03 |
| 3.00E+00 | 1.00E-15 | 9.00E+00 | 3.00E+01 | 7.34E-03 |
| 3.00E+00 | 1.00E-15 | 1.90E+01 | 3.00E+01 | 3.02E-02 |

Results with time dependent hazard rate. Offset establishes the point of the beginning of the annual calculations.

| BETA | LAMBDA | OFFSET (YEARS) | TEST INT (DAYS) | OSAVG |
|----------|----------|-------------------|--------------------|----------|
| 1.00E+00 | 5.76E-08 | 0.0 | 3.00E+01 | 2.05E-05 |
| 1.00E+00 | 5.18E-07 | 0.0 | 3.00E+01 | 1.84E-04 |
| 1.00E+00 | 1.44E-06 | 0.0 | 3.00E+01 | 5.12E-04 |
| 1.00E+00 | 2.82E-06 | 0.0 | 3.00E+01 | 1.00E-03 |
| 1.00E+00 | 4.66E-06 | 0.0 | 3.00E+01 | 1.66E-03 |
| 1.00E+00 | 6.96E-06 | 0.0 | 3.00E+01 | 2.47E-03 |
| 1.00E+00 | 9.37E-06 | 0.0 | 3.00E+01 | 3.32E-03 |
| 1.00E+00 | 1.29E-05 | 0.0 | 3.00E+01 | 4.57E-03 |
| 1.00E+00 | 1.66E-05 | 0.0 | 3.00E+01 | 5.87E-03 |
| 1.00E+00 | 2.08E-05 | 0.0 | 3.00E+01 | 7.35E-03 |
| 1.00E+00 | 8.75E-05 | 0.0 | 3.00E+01 | 3.02E-02 |

Results with a constant ($\beta=1$) hazard rate. The value used is determined by evaluating the time dependent hazard rate at mid year of the annual calculation.

Table 4.4. Comparison of Calculations Using the Old-Old Renewal Option and a $\beta=3$ Hazard Rate With Those Using an Equivalent Constant Failure Rate. The agreement is excellent after the second year.

the two methods of calculating the annual average unavailabilities.

For this type of application the primary advantage of the Old-Old renewal option is its ability to propagate a gradual change in the failure rate throughout the entire lifetime of the component. The offset time automatically updates the failure rate to the proper value for the year of the calculation. Because it does not change by much during an individual test interval, the increasing failure rate can be approximated by a constant failure rate calculated at the midpoint of the calculation interval, but this would require manual updating for each new year.

Maintenance Policies

The Old-Old renewal option can be used to investigate maintenance policies in which the hazard rate is set back to its initial value by the maintenance action. This can be done by limiting the calculation time to the maintenance interval. With the Old-Old renewal option the effects of varying the maintenance (or replacement period) can be investigated by changing the calculation time. Figure 4.8 is an example of what the hazard rate resulting from the choice of two different maintenance intervals might look like.

Because the components of a complex system may be maintained at different intervals, with hazard rates being reset at throughout any particular period, this method of simulat-

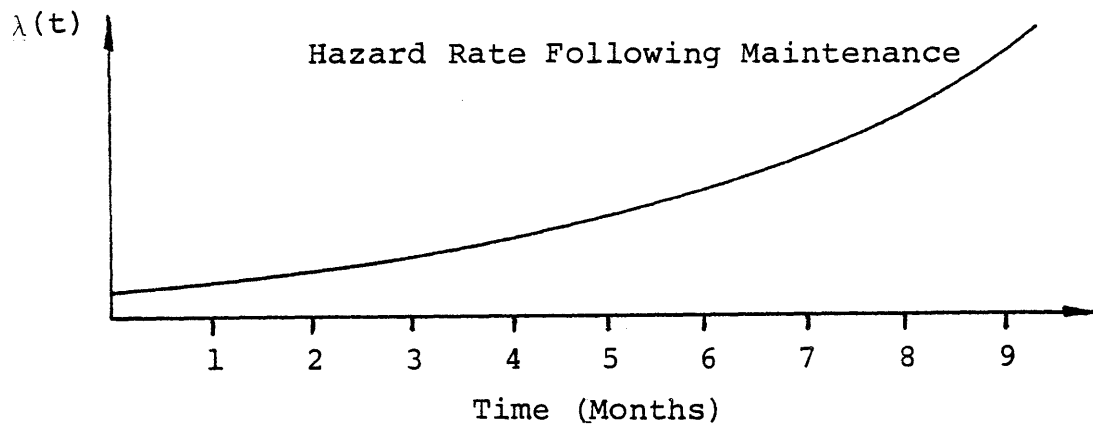
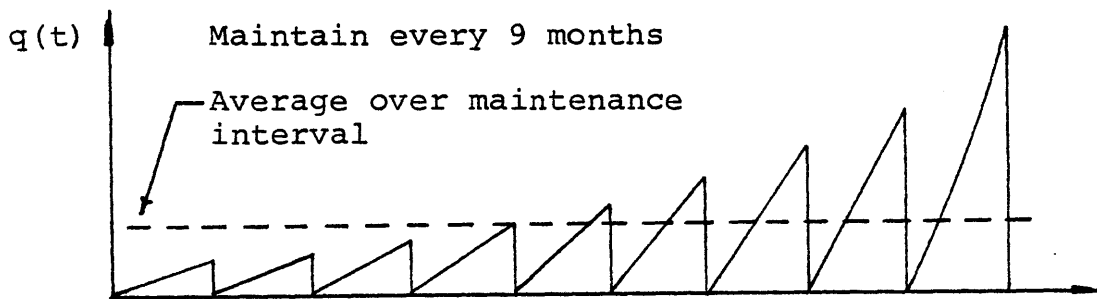
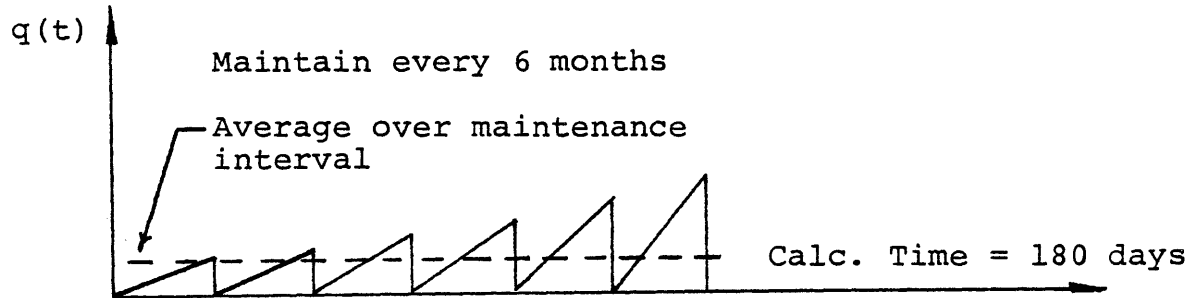
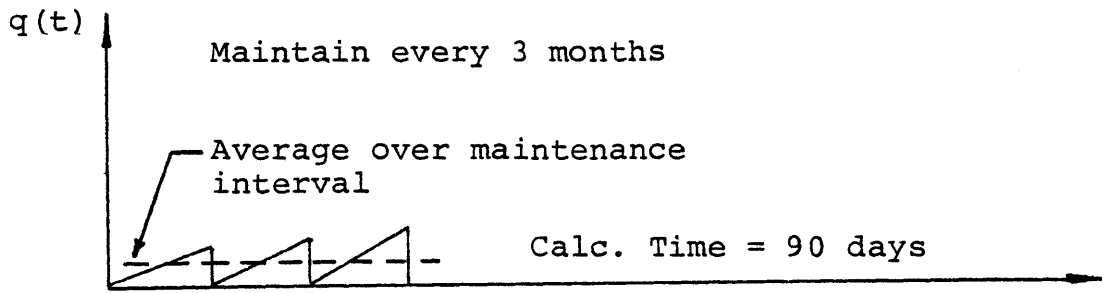


Figure 4.8. Effect of Variations in the Maintenance Interval on a Time Dependent Hazard Rate. Test monthly, maintain as indicated.

ing maintenance activities is not useful in multiple component systems. It is recommended that a provision be added to the code to reset the hazard rate after a user specified number of tests. This should not be a difficult option to add, as the computational routines currently count periodic tests during the calculation.

4.5 TEST CAUSED WEAR-OUT AND BURN-IN

The original FRANTIC II code provided for only the standby failure rate change. In the FRANTIC II-MIT Code both the standby and demand failure rates can be changed by tests. This modification makes the code more consistent, because tests can change a component's susceptibility to demand as well as standby failure mechanisms. The equations for test caused changes were presented in Section 3.5.3 and the idea of susceptibility was discussed in section 3.6.1. Here only their effects on the code's calculations will be addressed.

4.5.1 EFFECT ON STANDBY FAILURE RATE

The effect of test caused wear-out is to slightly increase the rate at which the component's unavailability increases during the next standby interval. Since the effect is cumulative in FRANTIC II-MIT, the more the component is tested the higher its failure rate will be at some later point in time. With the Old-Old renewal option, the offset time, t_0 , allows one to project the effect of testing

into the future to see the effect of test caused wearout over the design life of the plant. Figure 4.9 is an example of this projection.

Figure 4.9 is a calculation of the unavailability of a component which fails during standby due to a constant ($\beta=1$) failure rate, but is subjected to test caused wearout. The wear-out is modeled by increasing the scale factor by a factor of $f_{\lambda}=1.01$ at each test. For purposes of illustration it is assumed that the test does not cause any Effective Downtime as defined in Section 4.3.2. Five curves are shown in the figure. The first and lowest corresponds to the component's unavailability during the initial stages of plant life. Because there is no immediate penalty for tested (EDT=0 hr/test), the lowest unavailability is obtained when the component is tested often.

The next four curves are calculated over the first 20 test periods after the plant has been in service for the time indicated.⁵ These curves show the long term effects of test caused wear-out. At the lower test intervals, the component is gradually worn out by testing, so that it begins failing very often. It can be seen that an "optimum" test interval appears to exist at a 30 day test interval for the 10 year

⁵ When the OO renewal option is used in FRANTIC II-MIT, the Offset Time automatically accumulates test caused changes in hazard rate over that period. This option makes the curves easy to generate. See Section 3.5.3 and Appendix I.

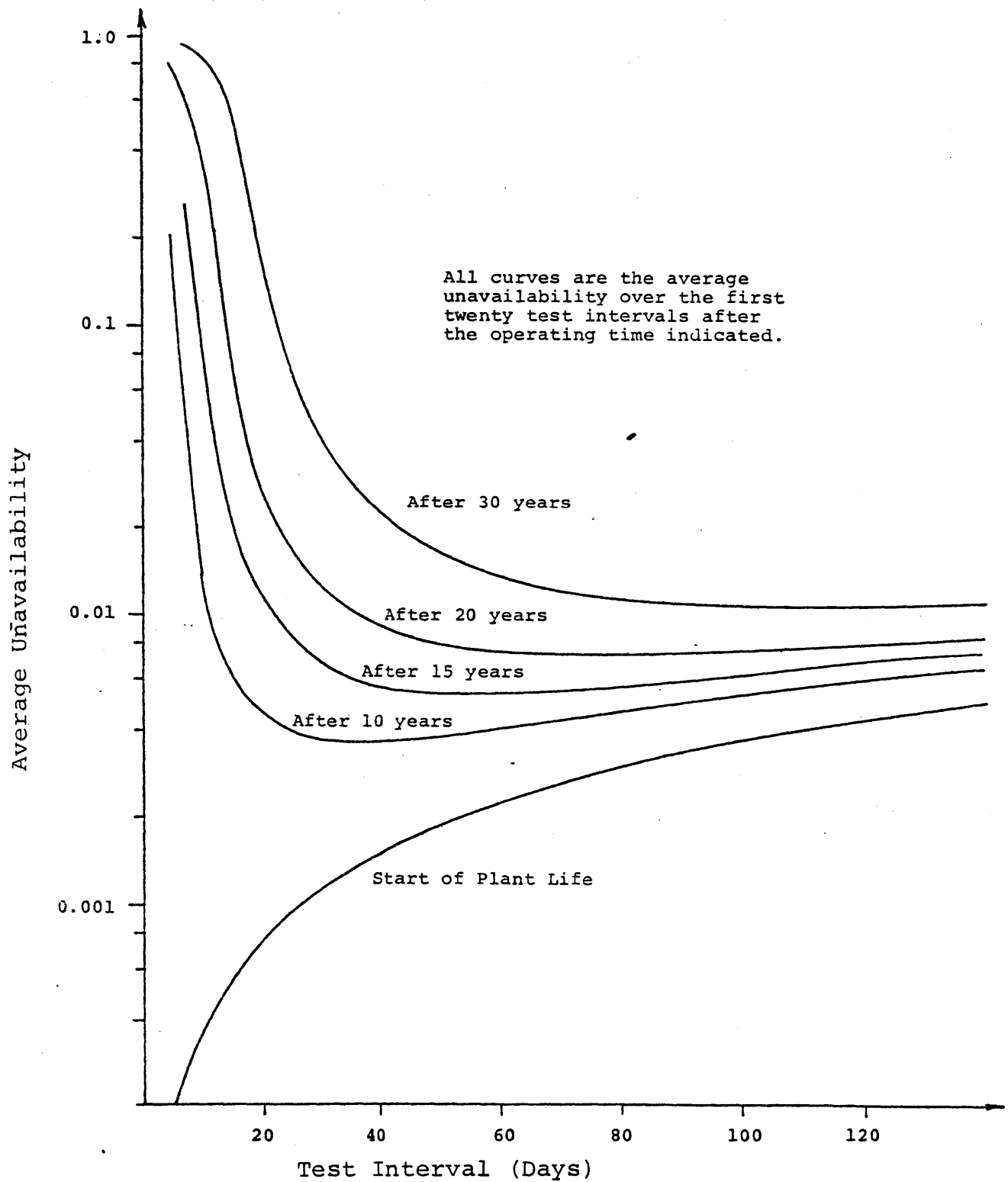


Figure 4.9. Long Term Effect of Test Caused Wear-out in Standby Failure Mechanisms. Example for $f_{\lambda} = 1.01$.

curve. This optimum goes to longer test intervals and becomes less sensitive to the test interval later in the plant life. Because of the storage dimensioning and the coding of the time point generating routine, a 30 year calculation would require recoding, so a calculation over the entire 30 years was not accomplished. However, if it were done it is believed that it would be very close to the 15 year curve.

The so called "optimum" test interval for the 20 test periods after 10 years appears to be the most reasonable interval to use if one is planning to overhaul the component on a ten year cycle. The criteria that would drive the decision would be whether or not the average unavailability of $3.5E-3$ is acceptable just before overhaul. The unavailability prior to that time will always be less, because fewer tests have been accomplished and the standby failure rate will be smaller. If a higher unavailability is acceptable before overhaul is necessary, the component should be tested on a longer interval to reduce the rate at which the failure rate increases. This will produce a higher initial unavailability, but one that is still lower than the target. A graph such as that given in Figure 4.6 would be useful to determine the range of unavailabilities produced by testing at a specific interval over the given number of years.

4.5.2 EFFECT ON DEMAND FAILURE RATE

Test caused wear-out of demand failure mechanisms

produces the same effect as it did for standby mechanisms. Because q_d is increased by each test, a short test interval produces a larger increase in q_d over a given life time than a long test interval. However, since testing is done to detect standby failures, the importance of these increases depends on the relative magnitude of the demand failure rate to the standby failure rate. In practical situations if the demand failure rate will be affected by a test, the standby failure rate will probably also be changed. Space does not permit a detailed investigation of all the possibilities.

4.6 SUMMARY

This chapter has attempted to give the reader a feel for the manner in which the unavailability of a component can depend on the various failure parameters that can be modeled in FRANTIC II-MIT. Because of the flexibility of the code it is very difficult to address all the possible applications. For the application to the High Pressure Coolant Injection System of a Boiling Water Reactor discussed in Chapter 6 it turns out that the constant standby failure rate model is adequate to provide most of the information required to make recommendations on test intervals.

The generalized Weibull hazard rate makes the code capable of modeling a wide variety of time dependent failure rates. The renewal options available to reset the hazard functions do not have the same degree of flexibility. A

maintenance model which could reset the hazard function every n tests might be useful to account for periodic maintenance or scheduled replacement of worn-out parts.

CHAPTER 5

APPLICATION TO MULTIPLE COMPONENT SYSTEMS

This chapter discusses the application of FRANTIC II-MIT to systems having more than one component. The primary power of the FRANTIC codes is their ability to calculate the unavailability of a complex system. Section 3.1.2 discussed how the selection of time points for the calculation of instantaneous unavailability and the numerical integration of the instantaneous unavailability calculated at these time points enables one to obtain the average unavailability of any multicomponent system without the use of cumbersome analytical equations. The analysis of multiple component systems using FRANTIC II-MIT is limited by only the capability to model the system's failure in terms of the unavailability of the individual components in the system and the storage capacity of the code. (Currently it is dimensioned for up to 100 components.)

In order to more efficiently address multiple component systems, a cut set generator and evaluator has been interfaced with FRANTIC II-MIT to accomplish the functions of the user supplied SYSCOM subroutine. The package is named CUTSETS and is taken directly from UNRAC (UNReliability Analysis Code), developed at MIT by Karimi [Ka80] and based on BIT, an unpublished work by

Wolf. [Wo75] The basic features of CUTSETS are discussed briefly in this chapter.

Following the presentation of CUTSETS, some of the considerations which could influence the choice of the periodic testing policy for simple systems are discussed. Many of the characteristics of simple multiple component systems have been analyzed analytically in the literature. Some of these results are presented in the references discussed in Chapter 2. This chapter is not intended to duplicate that research. Rather it will show how some special problems which might not be easily modeled analytically can be addressed with the CUTSETS - FRANTIC II-MIT package.

The chapter concludes with a short discussion of an approach to analyzing the periodic testing programs of standby safety systems in operating reactors.

5.1 CUTSETS

CUTSETS uses a top down algorithm and the method of bit manipulation to generate minimal cut sets. To speed the calculation time the user can specify that only cut sets with a number of components equal to or less than a specified size be retained during the evaluation. Cut sets larger than that value are automatically discarded. A flow chart of the cut set generator is given in Figure 5.1.

CUTSETS can process both complement and basic events, giving it the capability to evaluate fault trees containing

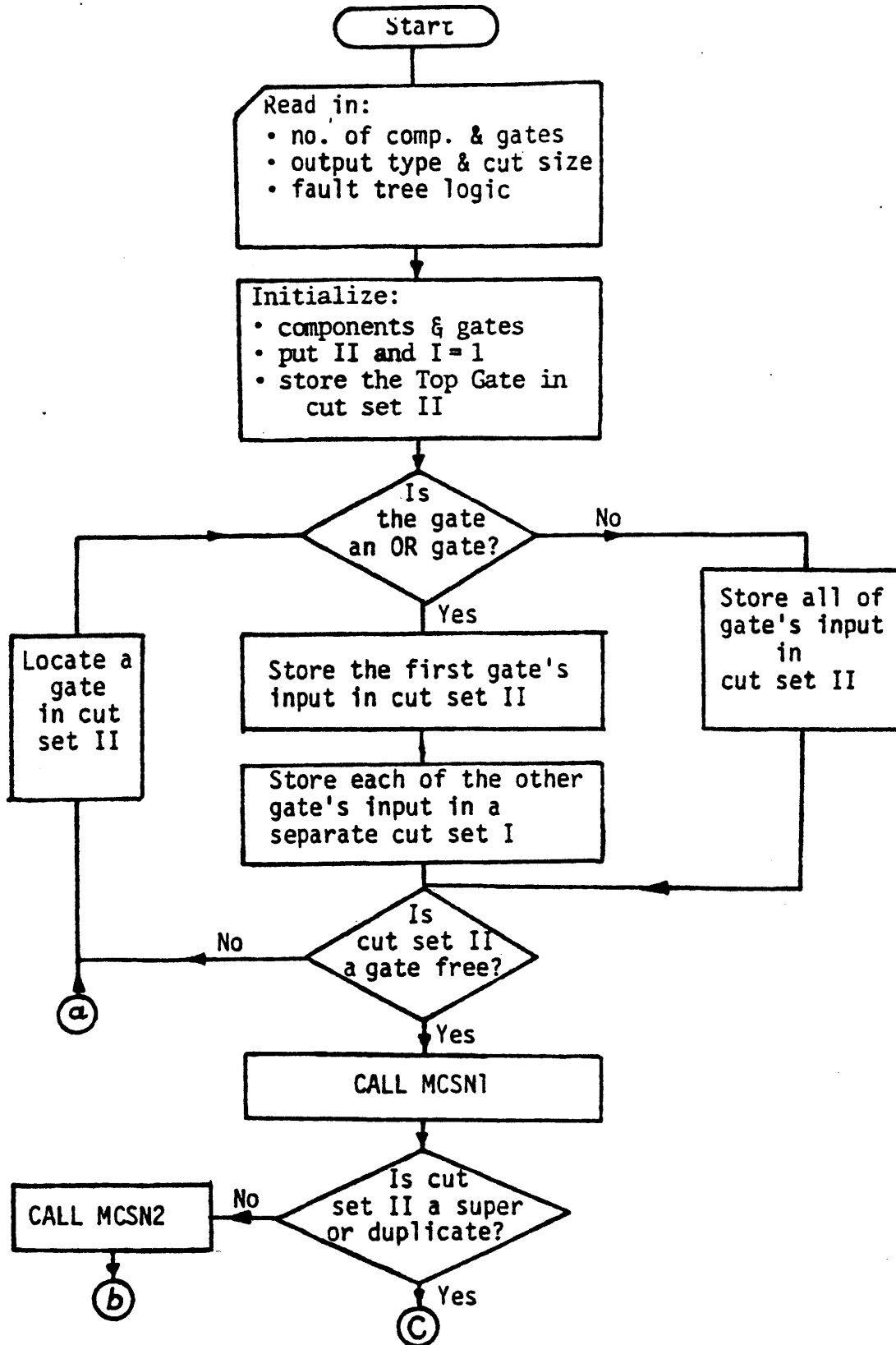


Figure 5.1. Cut Set Generator Flow Chart. [Ka80]

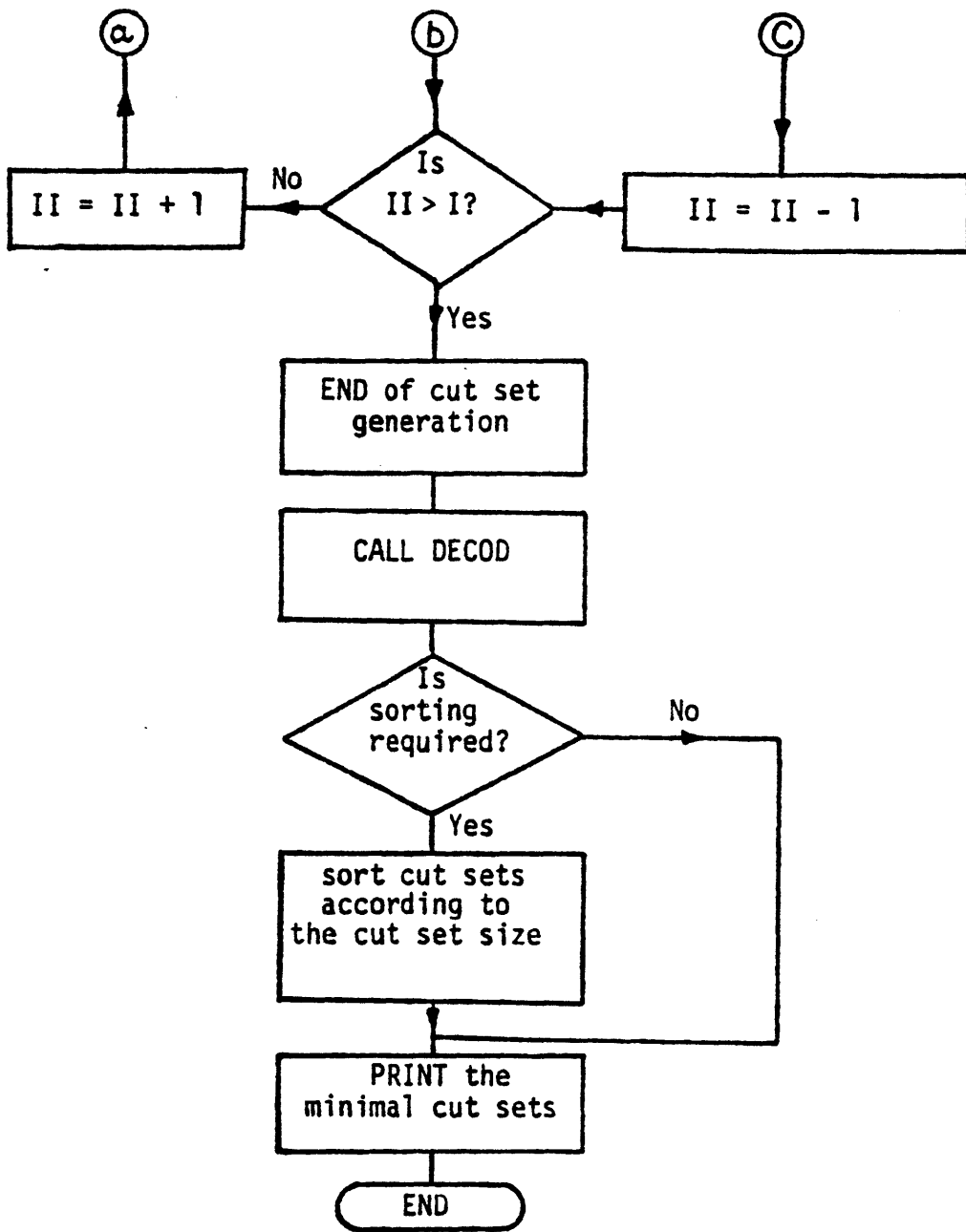


Figure 5.1. Cut Set Generator Flow Chart. (continued)

NAND, NOR, and Exclusive OR gates. However, since the cut set generator accepts only AND and OR gates, these special gates must be manually transformed before the logic can be input. Figure 5.2 illustrates how special logic gates can be modeled with AND, OR and NOT gates from basic events.

In adapting the cut set generator for use with FRANTIC II-MIT the following changes were made:

- The input format was modified to make more readable and easier to use.
- The maximum number of components per cut set was reduced from 30 to 10 to reduce storage requirements.
- FORTRAN statements were added to write the generated cutsets to a permanent file for repeated use.
- FRANTIC II-MIT was modified to read the appropriate cut set file and use the CUTSETS evaluation routine as the user supplied SYSCOM subroutine.

The unavailability of the Top Event is evaluated in terms of the individual component unavailabilites through the minimal cut sets in accordance with the following equation:

$$Q_{TOP} = \prod_{i=1}^N Q_i = \prod_{i=1}^N \prod_{j \in i} q_j \quad (5.1)$$

Where:

Q_i - Cut set unavailability

q_j - Component unavailability

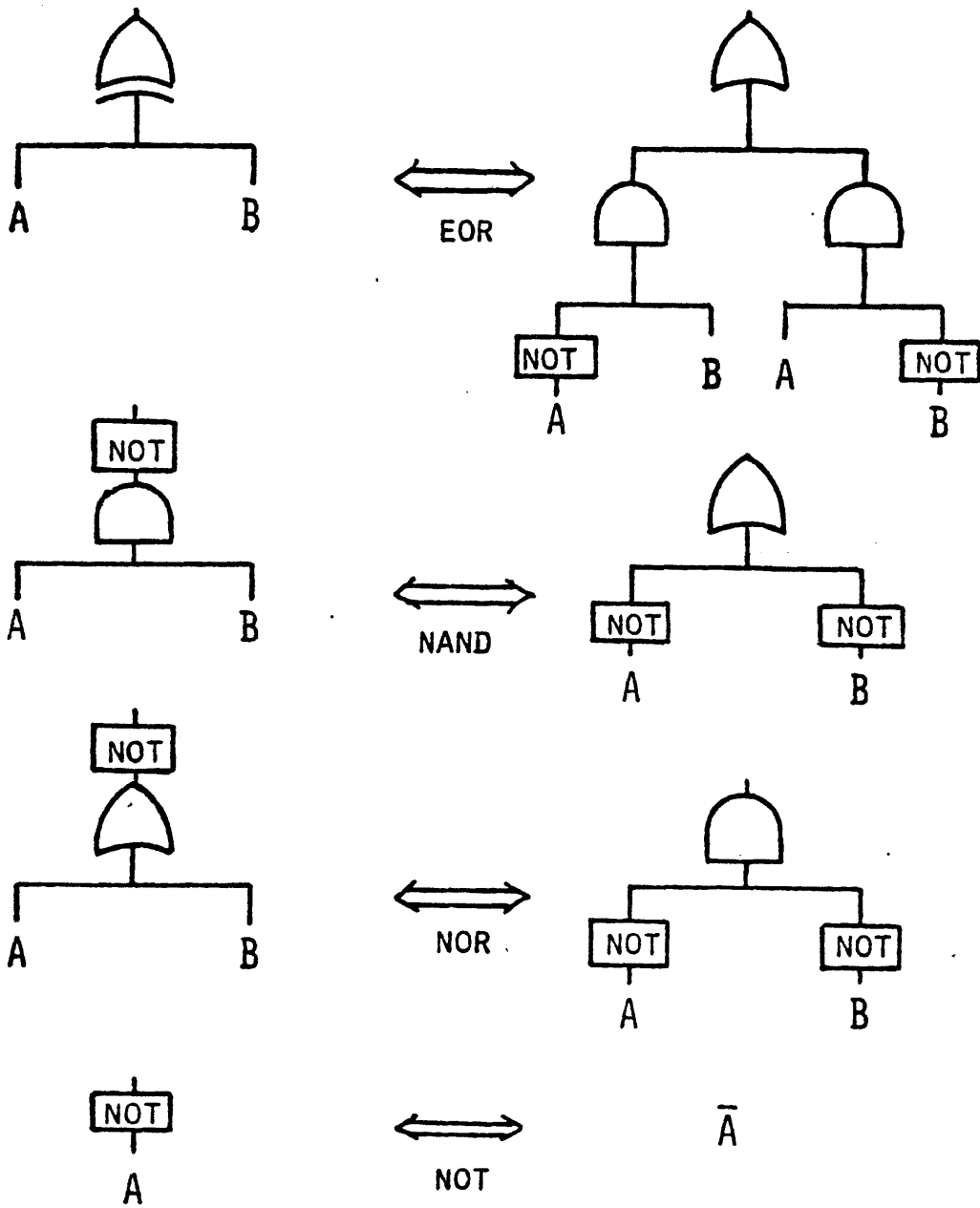


Figure 5.2. Equivalent Transformation of EOR, NAND, NOR, and NOT gates [Ka80]

N - Total number of cut sets

- Union of the minimal cut sets, which is by definition,

$$\prod_{i=1}^N Q_i = 1 - \prod_{i=1}^N (1-Q_i)$$

Expanding Equation (5.1) obtains the series:

$$Q_{TOP} = \sum_{i=1}^N Q_i - \sum_{\substack{i=1 \\ i < j}}^N Q_i Q_j + \sum_{\substack{i=1 \\ i < j < k}}^N Q_i Q_j Q_k - \dots + (-1)^N \prod_{i=1}^N Q_i \quad (5.2)$$

CUTSETS evaluates Equation (5.2) using the inclusion - exclusion principle. [He81, Ve70] The user has the option of evaluating the first term, the first three terms, or the first five terms of the expansion. Evaluation of only the first term is called the rare event approximation. It provides an upper bound estimate of the unavailability and runs much quicker on a computer than the second two options, which provide more precision. For most practical applications the rare event approximation yields sufficient accuracy if the analyst is careful not to assign high values of test caused unavailability to more than one series component. For further information on the algorithms that CUTSETS uses, refer to Karimi. [Ka80]

Input format and information on the use of CUTSETS on a VM/SP CMS computer are given in Appendix J.

5.2 SOME APPLICATIONS TO SIMPLE SYSTEMS

The discussions presented below are intended primarily to point out the flexibility available in applying FRANTIC II-MIT to systems and present some suggestions for using FRANTIC II-MIT. They are by no means complete, but are intended simply to illustrate the code's potential.

5.2.1 SERIES SYSTEMS

It is well known that series systems can be represented by a composite "super component" whose failure rate is equal to the sum of the individual failure rates. [He81] This can be used to combine the contributions of components which contribute to the Top Event through a common OR gate into one failure event. The major advantage is a decrease in the number of cut sets which must be manipulated by the code to obtain the system unavailability. The advantage can become large if the OR gate contributes to higher level AND gates.¹

If they are to be represented by a single failure event, the following conditions must be met: [Va79a]

- 1) All components in the series system are tested either simultaneously or successively, one after the other,
- 2) Repair is not started until all components have been tested, and

¹ The comparison with Vaurio in Section 5.3 is a good example. There the consolidation of failure events in three redundant pump legs reduces the number of cut sets from 8 to 1.

- 3) The components that need repair are repaired successively, one after another.

In terms of FRANTIC II-MIT, this last condition means that T_R is representative of the expected repair time of any combination of failures that might be observed during a test.²

The testing strategy for series components is straight forward. However, there are two points that should be mentioned.

- 1) The test interval should be determined considering the composite failure rate of all the components. The testing of the Turbine/Pump train in section 6.3 is done on the basis of all the components in the train.

- 2) If testing requires that the components be made unavailable to accomplish their safety function in the event of a true demand, testing of all the series components should be made at one time. If the components are maintained as separate failure events and the rare event approximation is used to combine the cut set contributions, care should be taken to account for the unavailability to override the test, q_0 , in only one of the series components. If this is not done, an over estimation of test caused unavailability

² Recall from Section 4.3.1 that the time required to repair a periodically test component is relatively minor compared to the time that the failure of the component might remain undetected.

will result, giving the potentially false impression that longer test intervals are warranted.

5.2.2 DEMAND FAILURES AND REDUNDANT COMPONENTS

When two components are in parallel each component has another which can accomplish the safety function while it is being tested. Figure 5.3 shows that for this type of system the optimum test interval depends on the demand failure rate as well as the standby failure rate. If there is no demand failure contribution to component unavailability, the average unavailability of the two component parallel system decreases as the test interval decreases, despite the fact that each component must be made completely unavailable to accomplish the test. When the interval becomes very small, one component is out of service for testing almost all of the time. However, there is a high probability that the second component will be available, since it just came off of test and has not had time to fail.

Demand failures put a lower limit on the unavailability of the components. When the possibility of a standby failure having occurred becomes small compared to the demand failure rate, the advantages of further decreasing the test interval becomes small, and lost redundancy due to testing becomes the dominant consideration. Consequently, the system unavailability goes through a minimum and increases for shorter test intervals.

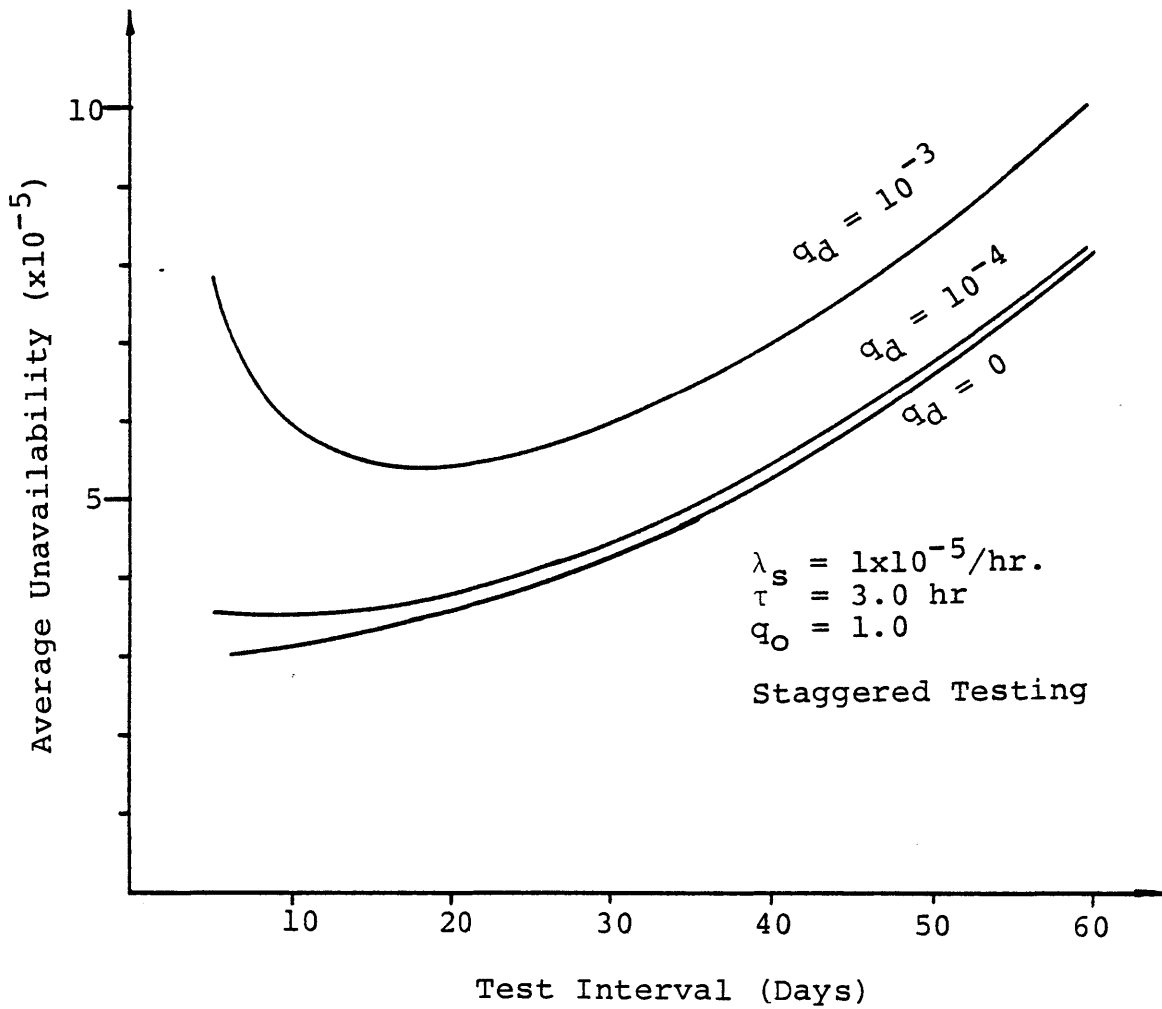


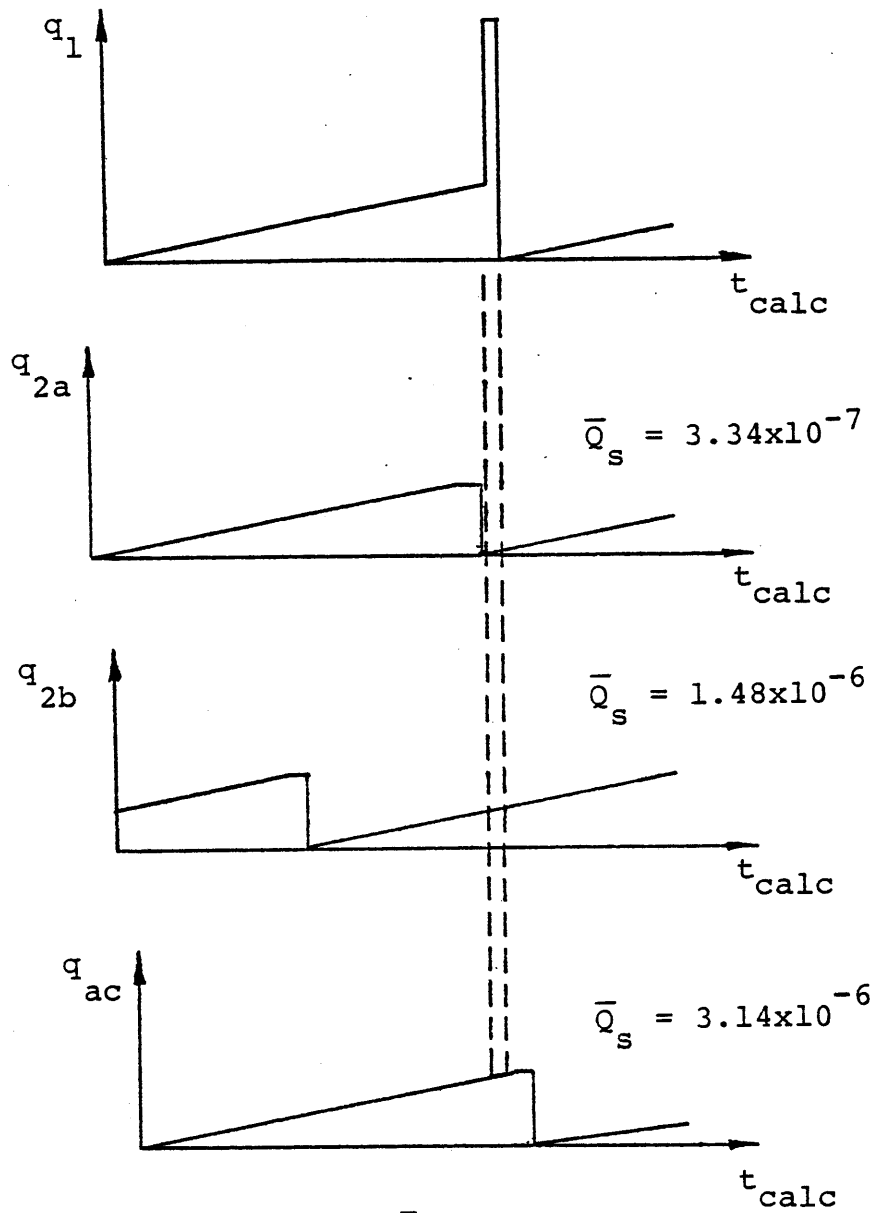
Figure 5.3. Effect of Demand Failure Mechanisms on the Optimum Test Interval of a Parallel Two Component System.

5.2.3 EFFECT OF UNEQUAL TEST OVERRIDE

Consider two different types of components which perform the same safety function. One component must be made unavailable for testing while the second is available during its test period. For this combination of components a sequential test policy provides lower unavailability than staggered testing. Figure 5.4 illustrates this. If the second component is tested just before the first, it will have a very low unavailability while the first component is being tested. Consequently, the unavailability created by the first component's test is minimized by the fact that the second component is in its most available condition. If the tests are staggered or accomplished in reverse order, there is a much higher probability that the second component may have failed since its last test. The resultant system unavailability is increased by a factor of 3 and 10 respectively.

5.3 COMPARISON WITH VAURIO'S 1-OUT-OF-3 SYSTEM CALCULATION

To illustrate the use of FRANTIC II-MIT for addressing system unavailability, a comparison with a calculation made by Vaurio and Sciaudone is made. [Va79b] Their work in modeling failure mechanisms of standby components was discussed in Chapter 2. They derived a code, ICARUS, to calculate the average unavailability of m-out-of-n redundant



$$\text{where: } \bar{Q}_s = \frac{1}{T_{\text{calc}}} \int_0^{T_{\text{calc}}} q_1(t) q_2(t) dt$$

Figure 5.4. Comparison of Test Policies for Parallel Components with Unequal Unavailabilities During the Test Period.

systems for up to $m=n=4$. Part of their work compared ICARUS with FRANTIC and found large discrepancies for sequentially tested components with more than one component per train. In validating the FRANTIC II-MIT - CUTSETS package, the calculations of Vaurio were repeated to determine the reason for the discrepancy. It was determined that the discrepancy results from rounding off T_1 , the parameter which staggers the tests, to nearest whole hour in subroutine COMPON.³ The parameter T_1 should not be rounded off, as sequentially tested components may be tested at very close intervals and inadvertant overlapping of tests of redundant components may result in large calculation errors.

This section uses Vaurio and Scaudione's test system to illustrate the application of the CUTSETS - FRANTIC II-MIT package to multiple component and shows results of the comparisons between the two codes both before and after the correction to FRANTIC II-MIT. It is shown below that the correction produces excellent agreement where there was once an order of magnitude discrepancy.

System Description and Cut Sets

The system in question is a 1-out-of-3 valve system

³ The error can be eliminated by changing
TEST1(NCOMP)=INT(TEST1(NCOMP)*24.0 + 0.5)
to:
TEST1(NCOMP)=TEST1(NCOMP)*24.0
This change must be accomplished three times.

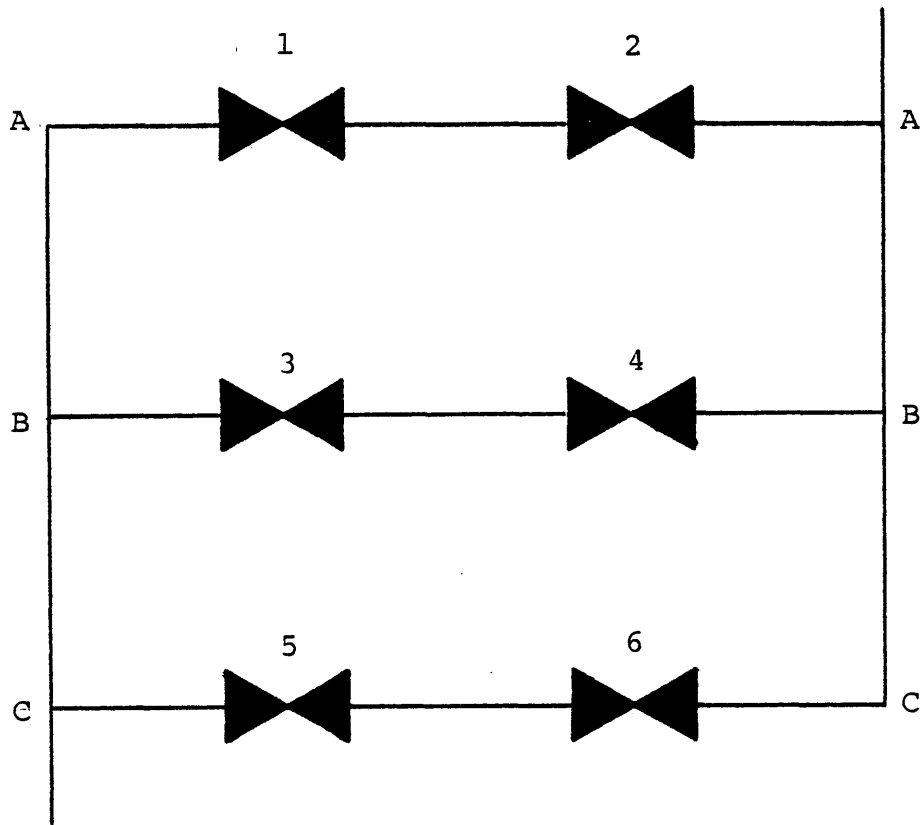


Figure 5.5. Single Line Diagram of the 1-out-of-3, Two Valve Per Redundancy, Test System Used in the Comparison Between FRANTIC and ICARUS.

with two valves per leg. Any one leg can accommodate the required flow, but both valves must open to obtain flow through any one leg. A line diagram of the system is given in Figure 5.5.

To use the FRANTIC II-MIT - CUTSETS package, we first draw the fault tree shown in Figure 5.6. Following the format given in Appendix J, we then create the logic file for input to CUTSETS shown in Table 5.1. The top line states that there are 6 components, a total of 10 components and gates, no more than 3 inputs to any one gate, sorting is desired (0), and the maximum allowable components per cut set is not more than 3. Each successive line gives the gate number, type gate (0 = OR, 1 = AND), and identity of input gates or components to that gate. The output from the code is given below the input.

To calculate the system's unavailability we establish an input file of component failure parameters from Table XVII of [Va79b], which is reproduced in Table 5.2. The resultant file is given in Table 5.3. Since T_1 is input in days, it is necessary to convert the 1.5 hours staggering interval into multiples of 0.0625 days. The cause of the discrepancy reported by Vaurio was in FRANTIC's conversion of T_1 back to hours.

Vaurio compared exact, ICARUS, and FRANTIC calculations. These results are given in Table 5.4 along with the results of calculations with both the uncorrected and cor-

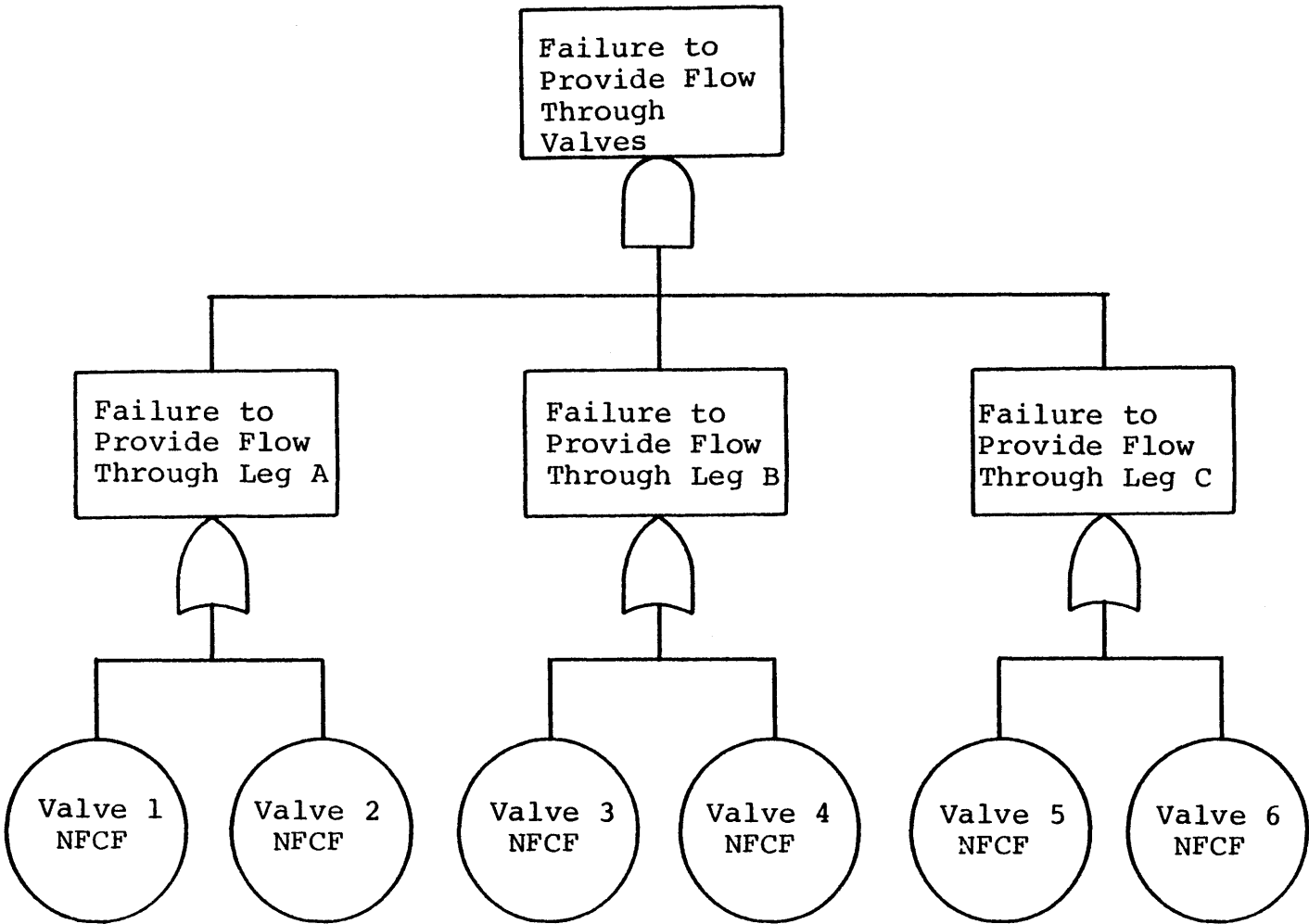


Figure 5.6. Fault Tree of the 1-out-of-3, Two Valves Per Redundancy, Test System.

FILE: VAURIO13 LOGIC A1

| | | | | |
|----|----|---|---|----|
| 6 | 10 | 3 | 0 | 3 |
| 7 | 1 | 8 | 9 | 10 |
| 8 | 0 | 1 | 2 | |
| 9 | 0 | 3 | 4 | |
| 10 | 0 | 5 | 6 | |

FILE: VAURIO13 CUTOUT A1

| 1 | TABLE -1 | FAULT TREE | LOGIC | | | |
|---|----------|------------|-------|-------|----|-------|
| 0 | GATE NO. | GATE TYPE | INPUT | COMP. | OR | GATES |
| | 7 | 1 | 8 | 9 | 10 | |
| | 8 | 0 | 1 | 2 | 0 | |
| | 9 | 0 | 3 | 4 | 0 | |
| | 10 | 0 | 5 | 6 | 0 | |

0 MINO IND LMIN II I

0 0 8 9 8

1 * TOTAL NUMBER OF CUT SET GENERATED = 8 *

0 TABLE - 2 ;
CUT SET NO. NO. OF COMP. IN C. S. COMPONENTS NOS.

| ----- | -- | -- |
|-------|----|-------|
| ----- | -- | -- |
| 1 | 3 | 1 3 5 |
| 2 | 3 | 2 3 5 |
| 3 | 3 | 1 4 5 |
| 4 | 3 | 1 3 6 |
| 5 | 3 | 2 4 5 |
| 6 | 3 | 2 3 6 |
| 7 | 3 | 1 4 6 |
| 8 | 3 | 2 4 6 |

*** TIME USED TO GENERATE THE MIN. CUT SETS WAS =.7999998E-01 SECONDS

Table 5.1. Input and Output of CUTSETS Application to the Test System.

| Parameter FRANTIC Notation | Exact Calculation | ICARUS Input* | FRANTIC Input |
|----------------------------------|------------------------|------------------------|------------------------|
| λ | $3.0 \times 10^{-7}/h$ | $6.0 \times 10^{-7}/h$ | $3.0 \times 10^{-7}/h$ |
| P_f | 0.001 | 0.001 | 0.001 |
| q_o | 0.10 | 0.10 | 0.10 |
| τ | 1.5 h | 3.0 h | 1.5 h |
| T_R | 10.0 h | 10.0 h | 10.0 h |
| T_2 | 720 h | 720 h | 30.0 d |

Table 5.2. Input Parameters for the 1-out-of-3, Two Valves Per Redundancy, Test System. *ICARUS represents the two series valves by an effective "single" component.

```

FILE: VAURIO3 DAT      A1

PRINT
-1
TITLE
VAURIO 1-OUT-OF-3 SYSTEM, 2 COMP PER LEG, DATA ON PAGE 56 OF VA79B
TIME
363.          0.5
COMP
NEW
 1 1          .3          30.          1.5 10.  0.1  .001
 1 2
 2 1          .3          30.  .06251.5 10.  0.1  .001
 2 2
 3 1          .3          30.  .12501.5 10.  0.1  .001
 3 2
 4 1          .3          30.  .18751.5 10.  0.1  .001
 4 2
 5 1          .3          30.  .25001.5 10.  0.1  .001
 5 2
 6 1          .3          30.  .31251.5 10.  0.1  .001
 6 2
-1
RUN
 1 TOTL NONE NONE          STAGGERED AT TEST INTERVAL, DEL TIME = 0.5 DAY
-1

```

Table 5.3. Input of FRANTIC II-MIT Application to the Test System.

| | Vaurio | FRANTIC-MIT |
|------------------------|------------------------|------------------------|
| Exact | 3.111×10^{-9} | -- |
| ICARUS | 3.145×10^{-9} | -- |
| Uncorrected FRANTIC | 2.310×10^{-8} | 2.36×10^{-8} |
| Corrected FRANTIC | --- | 3.165×10^{-9} |

Table 5.4. Results of Comparison of Uncorrected and Corrected Versions of FRANTIC with Vaurio's Calculations.

rected version of FRANTIC II-MIT. It can be seen from this table that good agreement is obtained between the corrected version of FRANTIC II-MIT and the exact and ICARUS calculations.

It can be seen from the above example that FRANTIC II-MIT can be used to analyze multiple component systems quite easily. Because the input for both the system's structure and the component failure characteristics are straight forward, the analyst has the flexibility to investigate the effects of changing either.

5.4 AN APPROACH FOR ANALYZING SYSTEMS

In this section the general procedure used to apply FRANTIC II-MIT to an operating standby safety system is outlined. It is not intended to be prescriptive. It follows the philosophy of NUREG-0492, which states:

System analysis is a directed process for the orderly and timely acquisition and investigation of specific system information pertinent to a given decision.

Accordingly, the approach outlined here is system oriented and directed towards providing information upon which to base periodic testing program decisions in an operational standby safety system. To accomplish this task, the following steps are followed:

- 1) Define the system's safety functions. Identify the conditions of the demand and the requirements of the response.

Determine how the system interacts with other systems which can also accomplish the safety function.

2) Determine how the system can fail to accomplish its safety functions. Establish the external boundaries of the system and interfaces with other systems. Based on the preliminary understanding of the safety functions and the system define the TOP Events of the safety function fault trees. There should be a fault tree for each safety function.

3) Identify the components in the system and the functions they perform. The limit of resolution should extend down to individual components that should be addressed in the testing procedure. Group components by the functions they contribute to and identify those which affect more than one safety function.

4) Construct fault trees for each safety function.

5) Correlate the fault tree failure events with the plant test procedures. Follow the procedures step by step to determine what component functions are actually tested. Determine if any components or functions are not tested by the procedures. Change procedures as necessary to include all components and functions in the procedures. Identify those components which are tested by more than one procedure. Compare conditions of the test with the expected conditions of a true demand. Identify any test created unavailabilities.

6) Quantitatively evaluate the testing policy. Test intervals should be varied at either a weekly or semi-monthly

interval. These intervals are practical because they are easy to schedule. Run the quantitative calculations over an operating cycle. Consider test prerequisites and manpower when establishing test staggering intervals. Subtrees that contribute directly to the TOP Event may be analyzed individually to reduce calculation time. Actual calculations should be based on a qualitative analysis of the fault tree cut sets and the test procedures.

7) Set up calculations to answer specific questions and judge the relative effectiveness of the various tests. The process can be iterative and involve minor modifications to the system as well as changes in the procedures or the interval at which they are accomplished.

CHAPTER 6

APPLICATION TO A HIGH PRESSURE COOLANT INJECTION SYSTEM

6.1 INTRODUCTION

The High Pressure Coolant Injection System will be the primary vehicle for illustrating the practical application of FRANTIC II-MIT to standby safety systems. On the surface the system is quite straightforward. Its major components are not redundant, so the primary contributors to its unavailability are rather obvious, single component cut sets which account for failures to make transitions during initiation. The system contains, however, many good illustrative examples of considerations that one must make in applying time dependent unavailability analysis to real systems. Currently, 21 separate procedures test various components and functions of the system. The resulting testing requirement is over 150 tests per year. Given the effort required to accomplish these tests, there is good reason to investigate the testing policy for the system from an unavailability point of view.

The approach to the analysis is to let the HPCI system analysis generate questions and find out if FRANTIC II-MIT can address them. The next section lists a variety of questions that have arisen. In every instance FRANTIC

II-MIT is flexible enough to address the problem and give valuable quantitative information to assist in making recommendations. The power of the code is not its ability to find the optimum test interval. It can not do that automatically. Rather, its primary advantage is the flexibility it gives the analyst to try a variety of approaches to improving system availability.

Organization of Analysis

First the safety functions and major components of the HPCI System are briefly described. The logic and actions necessary for the successful accomplishment of the safety functions are then presented. The assumptions used in deriving the fault trees for each safety function are then listed, and some of the features of the trees are discussed. (Fault trees, cut sets, and representative component failure data are contained in appendices to the chapter.) The analysis is broken into three major functional groups:

- 1) Turbine/pump train operability.
- 2) Initiation logic
- 3) Autoisolation logic and function.

The specific procedures which address each group are examined in detail, both qualitatively and quantitatively to:

- 1) Insure that they accomplish a complete verification of the functions they address. As necessary, specific procedural changes are recommended to address failure mech-

anisms that could be left undetected by the existing procedures.

2) Relate the procedures to fault tree failure events and quantitatively determine the impact changing the test interval on the unavailability of the system to perform its safety functions.

3) Investigate to effects of various periodic testing strategies on safety function unavailability and recommend changes to improve availability. (In one instance this includes a design modification on the system.)

The analysis has uncovered many good illustrative examples of FRANTIC II-MIT applications to the probabilistic risk analysis of real reactor systems. Among them are:

1) Consolidation of series components into super components with composite failure parameters (Section 6.4).

2) Importance of non fault tree components as contributors to test caused failures (Section 6.4).

3) Use of failure events to model the condition of the entire system being disabled to allow testing its initiation logic without interfering with normal reactor operations. (Section 6.5)

4) Calculations where a specific periodic test can affect more than one fault tree (Section 6.6).

5) The effect of design modifications to the system (Section 6.5).

6) Effect of common cause failures resulting from calibration of a group of sensors at one time (Section 6.5).

7) Effect of common cause failures due to the limited ability of specific groups of components to respond to a true demand (Section 6.6).

8) Effects of staggered testing of a set of parallel components in series with a second set of components (Sections 6.5 and 6.6).

It will be shown in this chapter that application of time dependent unavailability analysis reveals many areas where the HPCI system periodic testing policy can be improved.

6.2 DESCRIPTION OF SYSTEM FUNCTIONS

The High Pressure Coolant Injection (HPCI) System is a single leg, steam turbine powered pump and associated piping designed to provide up to 4,250 gpm of water to the reactor vessel via feedwater line "B." It operates over a pressure range of approximately 150 to 1,000 psig. Steam produced by decay heat is used to drive the turbine. The steam is taken from the main steam supply line upstream of the main isolation valves. The water supply to the pump is provided by the Condensate Storage Tank or the Suppression Pool. The

¹ The numbers 23 and 2301 appear throughout this chapter. They identify a component as belonging to the HPCI System.

system is designed to operate independently of AC power, with the exception of one (of two) autoisolation valves (MO 2301-4).¹ A simplified diagram of the HPCI System is given in Figure 6.1.

6.2.1 SAFETY FUNCTIONS

In the event of a loss of coolant accident the HPCI system must accomplish one of two safety functions:

1) Given the break has occurred elsewhere in the pressure boundary, the HPCI System must automatically deliver its rated output of water to the core upon demand.

2) Given the break has occurred within the HPCI steam supply line, the system must automatically isolate the break from the reactor.

In the manual or test mode the HPCI System is also used to provide cooling and/or controlled depressurization to the nuclear vessel in conjunction with the Automatic Depressurization System (ADS) during transients which isolate the primary containment. When decay heat generation has been reduced to about 2 % of full power (at approximately 20 minutes after shutdown), the HPCI System can provide this function without the use of the ADS. Water discharged from the pump can be routed back to the Condensate Storage Tank, with the HPCI being used primarily as an energy sink for decay heat steam, or a controlled amount of makeup water can be provided to the reactor by splitting flow between the injection line and the test return line using MO 2301-10.

HPCI System Diagram

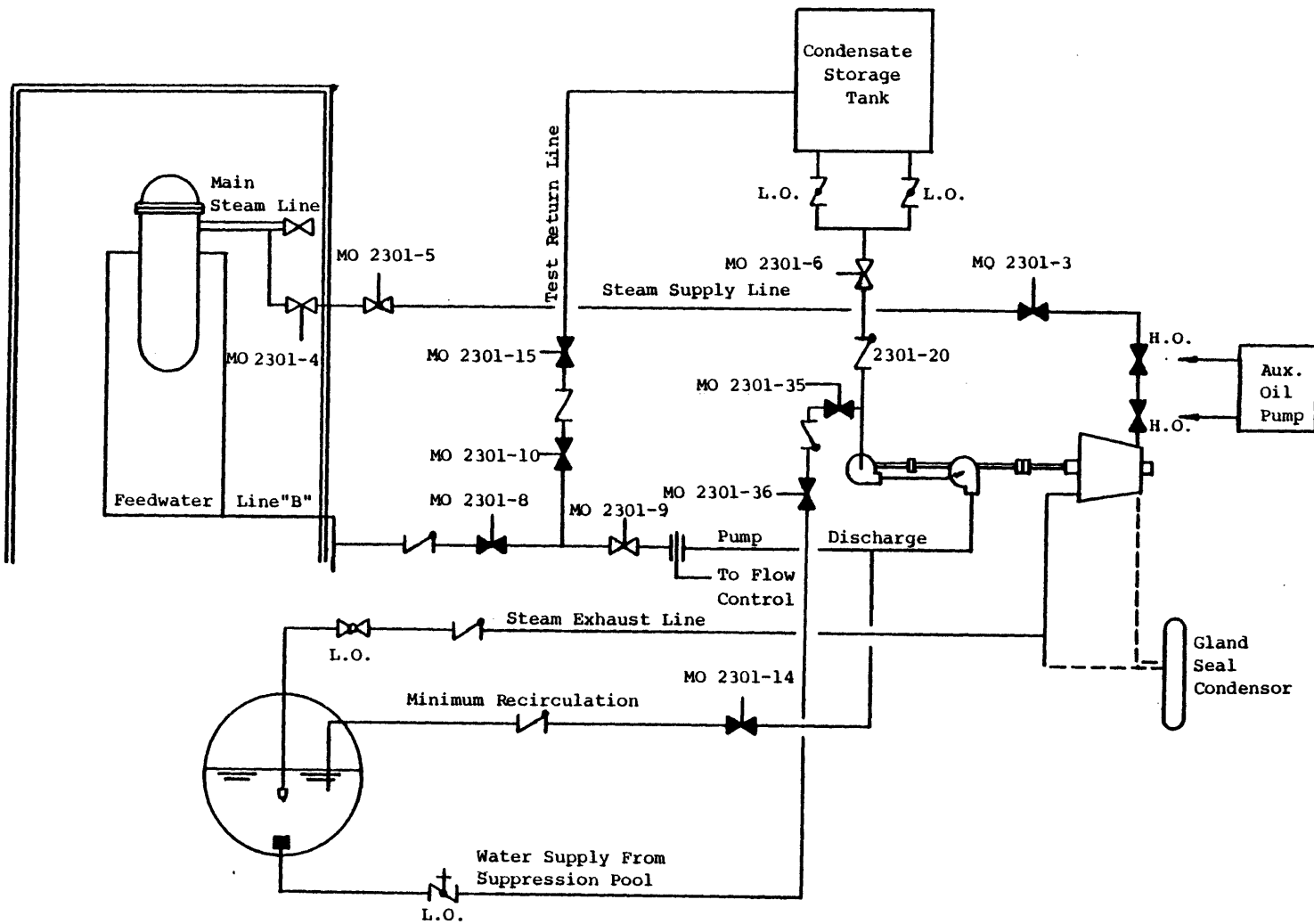


Figure 6.1. Simplified Diagram of the High Pressure Coolant Injection System of a Boiling Water Reactor.

6.2.2 INJECTION FUNCTION

HPCI System operation is initiated automatically by either low-low water level in the reactor vessel (approx. 49 inches below a reference water level or 78.5 inches above the active fuel) or high drywell pressure (approx. 2.5 psig), provided no autoisolation or turbine trip signal exists in the HPCI logic circuits. Four sensors monitor each parameter, each closing a switch when the setpoint is reached. (There is no external readout of the parameter.) The switches are arranged to complete a circuit when one-out-of-two, taken twice, logic is satisfied. The initiation signal produced by the sensors energizes a series of relays, which in turn close circuits that send signals to eleven different components in the HPCI system. Based on the analysis of the initiation logic tests, recommendations for changes in the design of the logic train are presented in Section 6.5 where a more complete description of the current design and recommended changes is given.

Actions Required for Injection

Steam required to power the turbine must pass through five valves before entering the turbine. The two autoisolation valves, MO 2301-4 and MO 2301-5, are normally open during reactor operation, and the steam supply line is at reactor pressure up to the Steam Supply Valve, MO 2301-3, which is normally closed. The turbine stop valve and control valve, which are hydraulically operated by the turbine

lubrication system, are also normally closed. Upon HPCI initiation, MO 2301-3 must open and the auxiliary oil pump must start and attain sufficient pressure to open the two hydraulically operated turbine valves.

Steam is exhausted from the turbine to the suppression pool through a 20 inch pipe containing a check valve and a normally locked open block valve. No active functions are required in the exhaust line upon initiation. However, the the check valve must not stick closed.

The HPCI turbine is mechanically coupled to a two stage water pump and booster. A flow controller monitors pump output and provides feedback to the turbine control valve. Upon initiation the controller set point must be correct and its instruments and logic circuits must be operating. Its failure could result in either too little water being injected into the core or a high speed turbine trip and loss of HPCI function entirely.

Water for the HPCI pump can come from two sources. The initial source is the Condensate Storage Tank. Water from this source passes through one of two normally locked open butterfly valves into a 16 inch pipe containing the normally open CST Supply Valve, MO 2301-6, and a check valve. The initiation signal opens MO 2301-6 in the event it is closed when the demand is made and the limit switches on the suppression pool isolation valves indicate that at least one valve is closed.

The source of water for the pump is automatically transferred to the suppression pool when:

- 1) Level switches in the CST open on low level (two-out-of-two logic).
- 2) Level switches in the suppression pool close on high level (one-out-of-two logic).

Relays in each of these circuits open MO 2301-35 and MO 2301-36. Limit switches in these two valves (two-out-of-two logic) then send a close signal to MO 2301-6 and an interlock prevents it from reopening.

Water supplied from the suppression pool to the pump goes through a strainer screen, a normally locked open butterfly valve, a check valve, and MO 2301-35 and MO 2310-36.

The HPCI pump discharges water into feedwater line B via a 14 inch pipe containing a normally open pump discharge valve, MO 2301-9, a normally closed pump discharge valve downstream of the test return line, MO 2301-8, and a testable check valve, 2301-7. There are two significant sources of water diversion:

- 1) A 10 inch test return line to the CST. This is normally blocked closed by two motor operated valves, unless a system operational test is underway.

- 2) A 4 inch minimum bypass line designed to protect the HPCI pump in the event the discharge line is blocked. Upon initiation the minimum flow bypass valve, MO 2301-14, opens

and then closes when sufficient flow is attained to verify that proper pump discharge is established. For conservatism it is assumed that MO 2301-14 must cycle properly to prevent diversion of water to the suppression pool in order for system success.

Once in feedwater line B, the water is prevented from flowing away from the reactor by a check valve and an isolation valve. It must pass through a check valve and a normally open isolation valve to reach the reactor. HPCI initiation has no control over any of these valves.

6.2.3 AUTOISOLATION AND TERMINATION

The HPCI injection function will be disabled for any one of the following reasons: 1) LOCA in the HPCI steam supply line, 2) Low reactor vessel pressure, 3) Turbine protection functions, or 4) High reactor vessel water level. The first three result in isolation of the HPCI System from the reactor vessel, with the consequent disabling of the injection function. The fourth results from two of the sensors that generate the low-low reactor water level signal and indicates that the injection function is no longer required.

LOCA in HPCI Steam Supply Line

The HPCI System is automatically isolated from the reactor and the turbine is tripped if a break or leak is detected in the HPCI steam lines. The autoisolation signal is produced by any one of four different groups of sensors.

Each has independent sets of sensors powered by both 125 VDC buses A and B to provide redundancy with respect to power supply. An autoisolation signal can be produced by one of the following:

- 1) Temperature of 170 degrees F in the torus room north west quadrant mezzanine behind rack 2257 in two-out-of-two temperature switches (one circuit on each bus).

- 2) Temperature of 170 degrees F in the Reactor building, north side above the HPCI valve station in two-out-of-two temperature switches (one circuit on each bus).

- 3) Temperature of 190-200 degrees F in the Turbine/Pump Room, west wall, elevation 31' in two-out-of-two temperature switches (one circuit on each bus).

- 4) High differential pressure of at least 180 inches H₂O (corresponding to 300% rated flow) across a 90 degrees turn in the HPCI steam supply line (one circuit on each bus). The autoisolation signal produced by any one of the above eight circuits will close the two steam supply isolation valves, trip the steam turbine, and inhibit both manual and automatic HPCI initiation until an operator manually resets the autoisolation seal-in on Panel 903 in the Control Room.

The eight leak detection circuits are not as redundant as they appear to be. They are located in different areas and may not all be able to see a leak that is occurring at one

specific location. For this reason they complement rather than duplicate each other. In the fault tree for the autoisolation function a conditional event is included which gives the probability that a given set of detectors can detect the leakage steam.

Low Reactor Vessel Pressure

The system will also autoisolate if reactor vessel steam pressure falls below the level at which it will no longer be sufficient to turn the turbine. The logic consists of four pressure switches (set point at 77 psig) connected in a two-out-of-one logic configuration. This circuit does not seal-in the autoisolation signal. If vessel pressure subsequently rises the HPCI may be reinitiated without a manual reset.

Turbine Protection Functions

A turbine trip without closure of the steam supply line isolation valves will occur when the following sensor switches are closed:

- 1) Low water pressure at pump suction (set point: 15 in. Hg, one-out-of-one logic).
- 2) High steam turbine exhaust pressure (set point: 150 psig, one-out-of-two logic).

No manual reset is required to enable the HPCI initiation circuit after a turbine trip due to turbine protection functions. The HPCI can be started either manually or by the

reoccurrence of vessel low-low water level or high drywell pressure provided the trip signal no longer exists.

High Reactor Water Level

HPCI System operation is terminated by a high water level signal (approx. 48 inches above the reference level) in both of the level switches used for this logic. The termination signal produces a turbine trip and a seal-in which must be manually reset before reinitiation is possible by any means other than a low-low reactor water signal.

Actions Required for Autoisolation

If an autoisolation signal is generated, one of two steam line isolation valves, MO 2301-4 or MO 2301-5, must close to isolate the steam line. To provide redundancy with respect to power sources, MO 2301-4 is operated by AC power while MO 2301-5 has a DC motor. In addition, the logic for each valve is powered by separate DC buses. Close signals are also sent to MO 2301-35 and MO 2301-36 in the event they are open. This isolates the suppression pool from the pump suction. The pump discharge valves are not closed by the autoisolation signal, but the testable check valve in the line can provide the necessary isolation. If the LOCA occurs in the HPCI system, the autoisolation signal will disable the HPCI initiation circuit, and all normally closed valves will remain closed.

6.3 FAULT TREE ANALYSIS

6.3.1 ASSUMPTIONS

The assumptions used in the fault tree are as follows:

1) The fault trees are developed down to the level at which individual components are periodically tested.

2) Individual component failure rates include:

- Failures of wires from the respective power bus to the component.

- Failures in sensor conduits or taps into process lines which would prevent a sensor from being exposed to the environment being measured.

3) The following faults are not considered:

- No water in the CST
- No water in the suppression pool

4) Failures of relays include failures of wires from the activating relay contacts to the control circuits of the operating equipment.

5) Common cause failures which occur at the time of a true demand are accounted for using a separate failure event and are modeled with q_d . This assumption makes their probability of occurrence unaffected by a periodic testing policy. Common cause failures modeled by q_d include:

- Design errors.
- Dependent failures.
- Extreme environments for which the sensors are not qualified.

- Human and calibration errors during sequential testing of redundant components. All sensors performing a given function are normally calibrated on the same month. Human error can result in a failure to recognize that the sensors are improperly calibrated when they put back into service.

Common cause failures due to calibration may also be modeled by a standby failure rate. In fact, calibration drift is a candidate for a time dependent hazard rate. Since all sensors of a given type are all calibrated during the same month, they all drift from their setpoints for the same period of time. There is normally a range in which the sensor can respond without hindering the effectiveness of the system, so there is a period during which the sensors have little chance of being far enough from the setpoint to degrade system performance. As the time since last calibration increases, the probability that the next small drift will cross the tolerance limit increases. This failure behavior can be modeled with a generalized Weibull hazard rate with a shape factor greater than 2. The conditional failure rate in this case would be increasing as the time since the last calibration increased.

6) No credit is taken for a manual initiation of the injection function during a small LOCA. It is a constant per demand probability and reduces the probability that the initiation function will fail. It should be noted that manual initiation of HPCI requires that the operator activate at

least four separate components, and in a high stress situation the probability for error can be quite high. However, this is offset by the fact that the operators manually initiate the HPCI System for the monthly Turbine/Pump Operability Test. The reliability of manual initiation could be increased by allowing the operator to energize autoisolation relays directly with one switch. However, this can increase the probability of inadvertent HPCI initiation without a LOCA present.

6.3.2 QUALITATIVE ANALYSIS

Because there are two safety functions which the system must satisfy, two fault trees are necessary to describe the system's safety unavailability.

A fault tree describing in detail the failures which can prevent HPCI injection upon demand as the system is currently designed is presented in Appendix A. The cut sets generated in the evaluation of this tree are given in Appendix B. This fault tree is dominated by single component cut sets, which for convenience are reproduced in Table 6.1. Revisions to the fault tree resulting from recommended design changes in the system's initiation logic are contained in Appendix D, and the resulting minimal cut sets appear in Appendix E. The design changes are discussed in detail in Section 6.5.2.

The autoisolation function fault tree is presented in Appendix F, with the resulting minimal cut sets appearing in

- 1 System down for repair of support equipment
- 2 Loss of 125 VDC Power from Bus D5
- 3 Loss of 125 VDC Power from Bus D8
- 4 Loss of 250 VDC Power from Bus D9
- 5 System unavailable due to initiation logic testing
- 12 23A-K23 or 23A-K24 (Initiation seal-in relays) Normally open, fails open
- 30 MOV 2301-3 (Steam to Turbine Valve) Normally closed, fails closed
- 31 MOV 2301-4 (Inboard Steam Supply Line Isolation Valve, AC operated), Normally open, fails closed
- 32 MOV 2301-5 (Outboard Steam Supply Line Isolation Valve, DC operated), Normally open, fails closed
- 33 Turbine driven pump failure
- 34 Steam turbine loss of function
- 35 Turbine lubrication system failure
- 36 HPCI Room cooler failed and required
- 37 LOCA in HPCI Steam Supply Line
- 39 2301-45 (Steam Discharge Check Valve) Stuck closed
- 40 2301-74 (Steam Discharge Manual Valve) Locked open, fails closed
- 41 Coolant discharge line rupture
- 42 AO 2301-7 (Air Operated Testable Check Valve) Fails stuck closed
- 43 MOV 2301-8 (Pump Discharge Valve from MOV 2301-9) Normally closed, fails closed
- 44 MOV 2301-9 (Pump Discharge Valve) Normally open, fails closed
- 45 MOV 2301-14 (Minimum Flow Bypass to Suppression Chamber) Normally open, fails open
- 46 Feedwater 57B line discharge isolation valve Normally open, fails closed
- 47 Feedwater 58B line discharge check valve Normally open, fails closed?
- 50 Human error probability: Failure to reset HPCI
- 51 Common cause failures in steam line low pressure sensors
- 52 Human error, common cause: miscalibration of high temperature sensors in steam line space
- 53 Human error, common cause: Miscalibration of turbine trip sensors, a) Pressure, 2) Level
- 54 23A-28 (Autoisolation initiation relay) Normally open, fails closed
- 55 False signal indicates turbine overspeed
- 56 PSL 2360 (Pump Suction Low Pressure) False signal indicating low pressure caused by contacts failing shorted

Table 6.1. HPCI Injection Function Single Component Cut Sets.

- 57 23A-K17 (Relays pump suction low pressure to turbine trip relay) False signal caused by contacts failing shorted
- 62 dPIS 2352 or 2353 (Steam Line Differential Pressure Sensor) False signal indicating
 - 1) Low range contacts failed shorted
 - 2) High range contacts failed shorted
 - 3) Human error: calibration
 - 4) Transient steam flow
- 63 23A-K9/K36 (Relays from Differential Pressure Sensors to Autoisolation Circuit), primary, calibration and common cause failures
- 70 High turbine steam exhaust pressure false signal, due to PSh 2368A pressure switch: Contacts fail shorted
PSh 2368B pressure switch: Contacts fail shorted
(Note: These sensors are not tested directly.)
- 72 23A-K12 (Relay from Steam Line Pressure Sensors to turbine trip circuit) Contacts fail shorted
- 75 23A-K6/K34 (Relays Turbine/Pump Room temperature sensors to autoisolation) circuit) primary, common cause, and calibration failures, Normally open, fails closed
- 76 23A-K8/K35 NOFC (Relays from valve station above 23 feet and torus compartment temperature sensors to autoisolation circuit) primary, common cause, and calibration failures
- 77 23A-K20 (Relay indicating high turbine exhaust) fails shorted

Table 6.1. HPCI Injection Function Single Component Cut Sets (continued).

Appendix G. Because of the apparent redundance of the steam line break sensors, the tree is layered with failure events that account for potential common cause failures. Of these, events 1, 10, 16, and 22 (all of which account for the inability of the dP sensors or temperature sensors at a specific location to detect the break) are judged to dominate the contribution of common cause failures. There are no single component cut sets for this tree and only two important double cut sets:

29, 31 - Both isolation valves fail to close upon demand.

19, 21 - Both Bus A and Bus B autoisolation relays fail to energize.

It will be shown in Section 6.6 that these cut sets dominate the quantitative analysis of the autoisolation function testing policy.

Because of the dual nature of the system's safety functions, the fault trees interact in two ways:

1) Because the autoisolation signal can override or terminate HPCI operation, testing to insure that the system can autoisolate reduces the availability of the system to perform its injection function.

2) Both of the Steam Line Isolation valves close as a result of the autoisolation tests. Valve failures during this test generate a requirement for repairs which make the valves unavailable to remain open for the injection func-

tion. This is accounted for by modeling test caused failures in Failure Events 31 and 32 of the injection function fault tree which follow the periodic testing interval of the autoisolation functional tests.

6.3.3 QUANTITATIVE ANALYSIS

Sections 6.4 through 6.6 contain a detailed discussion of the quantitative analysis of the fault trees. The analysis revolves about those calculations necessary to make recommendations regarding a periodic testing policy. As such, the relative change in the unavailability resulting from variations in the testing policy is more important than its absolute value, which is subject to the uncertainties of failure rate data.

Appendices C and H contain assessments of component failure rates used to determine the recommended periodic testing intervals discussed in Sections 6.4 through 6.6. These failure rates were derived from a number of sources. [WASH1400, N2232, GE80, IEEE-500, Plant data] Where the uncertainty in failure rates is considered important, the sensitivity of recommendations to variations in the failure rates of components is investigated.

6.4 TURBINE/PUMP TRAIN OPERABILITY TESTS

6.4.1 DESCRIPTION OF ONLINE TESTS

Currently, the injection function is tested by monthly

operational tests of the turbine/pump and cycling of all valves, with a few surveillance requirements added quarterly. Two monthly procedures are listed for the flow test at 1000 psig. A third requires quarterly surveillance in conjunction with the flow test. Finally, a fourth exercises the injection valves monthly. The procedures² are:

Procedure 8.5.4.1. HPCI Pump Operability and Flow Rate Test at 1000 psig. - The HPCI System is manually started, but then allowed to come to operational flow capacity under the influence of the flow controller. The flow test is accomplished using the Condensate Storage Tank (CST) as the water source. The water is pumped back to the CST through a full flow test bypass line. One of the test return line valves is preset to a partially open position to create a flow restriction so that the system's capability to pump its rated flow of water against the rated pressure head, as measured at the output of the pump, is verified. In addition to the flow test, several valves are cycled to verify their operability.

Procedure 8.1.6. HPCI System Pump and Valve Operability Surveillance - This procedure is the same as 8.5.4.1, with the exception that fewer valves are cycled and none are timed. It also requires the recording of selected hydraulic and mechanical data quarterly.

² Procedure numbers are from the plant's Operating Manual.

Procedure 8.A.15. HPCI System Integrity Surveillance

- A physical inspection of the HPCI piping is conducted quarterly in conjunction with the full flow test.

Procedure 8.5.4.4. HPCI Valve Operability Test - This test cycles the two steam supply line isolation valves (MO 2301-4 and -5) and the two pump discharge injection valves (MO 2301-8 and -9).

Recommendations for Consolidation

It is recommended that the first three procedures be combined into one. Procedure 8.I.6 is very nearly redundant with 8.5.4.1. The quarterly requirements of Procedures 8.I.6 and 8.A.15 are not great and could be easily added to 8.5.4.1 without making it too long. The one test could comply with all testing requirements, reduce paperwork, and make the actual testing policy clearer.

The requirement to cycle MO 2301-4 and 5 as part of 8.5.4.4 is redundant with the autoisolation functional tests, which also cycle these valves. It is recommended that the requirement to cycle those valves be dropped from this test. The resultant Procedure 8.5.4.4 involves the cycling of only one valve, which could be easily incorporated into 8.5.4.1. They are both accomplished at the same frequency and are subject to same Limiting Conditions for Operation test requirements.

6.4.2 DESCRIPTION OF OPERATING CYCLE TESTS

Three test procedures are currently done on a once per

refueling cycle basis. They are accomplished to verify functions that are not addressed by the monthly operational tests. Therefore, the failure mechanisms revealed by these tests are modeled as undetectable failure rates in components subject to more frequent testing. The procedures are:

Procedure 8.5.4.3. HPCI Pump Operability and Flow Rate

Test at 150 psig - This procedure tests the operability of the same components as 8.5.4.1, but at the lowest pressure that the system is designed to operate. This test is designed to reveal failure mechanisms that prevent the HPCI System from performing its function at lower reactor vessel pressures. For the purposes of defining undetectable failures in 8.5.4.1 revealed by this test, it is assumed that only the turbine is subject to low steam pressure failure mechanisms.

Procedure 8.5.4.6. HPCI Pump and Valve Operability

From Alternate Shutdown Station - This test is similar to 8.5.4.1, with the exception that manual initiation is accomplished from the Alternate Shutdown Station. It will not be addressed in this analysis for the following reasons: 1) Failure mechanisms which would be detected by this test which would not be revealed by the normal operational test are wiring and switch faults, which have a low probability of occurring, 2) Startup from the alternate shutdown station is primarily a manual requirement, which is redundant with the automatic initiation logic, and 3) Preliminary analysis

indicates that failures on nonredundant active components dominates the system unavailability.

Procedure 8.E.23. HPCI Instrument Calibration - The flow controller instrumentation is isolated from the system and checked with a pneumatic calibrator. Both as found and as left data are recorded. This test reveals possible degradations in the flow controller not observed in the monthly operational tests. Since the monthly operational tests will detect failures of the flow controller it is assumed here that the primary purpose of this test is to maintain the unit and prevent an increase in the hazard rate. The test is not addressed further in this analysis.

6.4.3 QUANTITATIVE ANALYSIS OF OPERATIONAL TESTS

The HPCI Turbine/Pump test satisfies the necessary conditions to be analyzed as a single "super component." It tests at the same time a total of 15 single component cut sets: 30, 33, 34, 35, 38, 39, 40, 41, 43, 44, 45, 50, 51, 55, 62. Repairs are not accomplished until the completion of a test and the system is declared inoperable until all repairs are completed.

Super Component Failure Rate

For the purposes of establishing a periodic testing policy, it is more reasonable to estimate a range of failure rates for the supercomponent than attempt to make a single point estimate. First the overall assessment is presented. This is followed by a discussion of the failure rate of the

Steam Supply Valve, which has been the major contributor to the unavailability of the HPCI System.

Table 6.2 lists failure data for the individual events which are accounted for by the super component. Five different sources are used to make the estimate. Events 50 and 51, corresponding to human error disabling the HPCI or transient steam causing autoisolation, are purely demand related. They occur only at the time of the true demand. Except for a few descriptive failure reports which give clues to possible failure mechanisms, there is no available breakdown of data into standby failure mechanisms verses demand generated mechanisms. In order to establish bounds for the standby failure rate, we assume that the failure data which does not distinguish between demand and standby contributions will contain some fraction of demand caused failures. First all data sources (with the exception of the two known demand failure mechanisms) are converted into a standby failure rate assuming demand failure rates correspond to a 30 day demand interval. Then the value of λ is reduced to $.75\lambda$ and a demand failure rate is calculated using $.25\lambda$ based on the 30 day demand interval. This procedure is repeated for a 50% division of failure mechanisms. The range of the composite failure rate data is summarized in Table 6.3.

The turbine/pump unit deserves special mention. It is represented by three failure events on the fault tree, but

Table 6.2. Assessed Range of Single Component Cut Sets Tested by the HPCI Turbine/Pump Test.

| Comp | WASH 1400 | GE Estimate | NPRD | F&MR, LER | Assessed Range |
|--|---------------|-------------|-------------|-------------|---|
| 30) MO 2301-3 | 0.3-3.0E-3/d | 1.6E-6/hr | 1.53E-6/hr | See text | Lambda = 5-25E-6/hr & qd = 0.001/d See total See total See total 10-26E-6/hr |
| 33) Turb parts | med=4.5E-3/d | | 11.5E-6/hr | 24.0E-6/hr | |
| 34) Turb pumps | | | No failures | No failures | |
| 35) Lub pump | 0.3-3.0E-3/d | | 1.3E-6/hr | No failures | |
| Total 33,34,35 | | | | | |
| 38) MO 2301-6 | 0.3-3.0E-4/d | | | | .08-0.8E-6/hr |
| 39) CV 2301-45 | 0.3-3.0E-4/d | 1.5E-7/hr | 5.0E-8/hr | No failures | 0.1-1.0E-6/hr |
| 40) LO 2301-74 | 0.3-3.0E-4/d | | | | 0.1-0.3/hr |
| 41) Pipe rupt | 1E-10/hr(10) | | | | Negligible |
| 43) MO 2301-8 | 0.3-3.0E-3/d | 1.5E-6/hr | 1.6E-6/hr | 3.0E-6/hr | 0.8-8.0E-6/hr |
| 44) MO 2301-9 | 0.3-3.0E-4/d | 0.15E-6/hr | 0.13E-6 | No failures | .08-.8E-6/hr |
| 45) MO 2301-14 | | | | | 0.1-1.0E-6/hr |
| 50) HEP | | | | | Est: 0.0001/d |
| 51) Trans Stm | | | | | Est: 0.0001/d |
| 55) Turb trip - false sig | 0.3-3.0E-7/hr | | | | 0.3-3.0E-7/hr |
| 62) dP False Sig | .01-1.0E-6/hr | | | | .01-1.0E-6/hr |
| Assessed range of Standby Failure Rate = 17 to 67 E-6 per hour | | | | | |
| Demand Failure Rate = 0.0012 per demand | | | | | |

the entire assembly functions as a unit. As it is a specialized piece of equipment, generic failure rates for the individual parts of the unit may overestimate its failure potential because they may double count certain failure modes.

It can be seen in Table 6.3 that the standby failure rate of the turbine/pump train super component is probably no higher than $67E-6/hr$. Considering that failure reports and occurrence rates verse standby time indicate that demand failure mechanisms do exist, the upper bound is taken to be $\lambda = 50E-6/hr$ and $q_d = .0075/d$. The lower bound is estimated to be $10E-6/hr$ and $q_d = .005/d$.

Steam Supply Valve Failure Rate

The Steam Supply Valve, MO 2301-3, is unique in that it is part of the reactor pressure boundary during the normal operation of the plant. It is subjected to much more severe environmental stresses than any other valve in the HPCI sys-

| | Low | | Geometric Mean | | High | |
|------------|-----------|-------|----------------|-------|-----------|-------|
| | λ | q_d | λ | q_d | λ | q_d |
| Increasing | 18 | .0012 | 34 | .0012 | 67 | .0012 |
| Demand | 13 | .0028 | 24 | .0044 | 50 | .0072 |
| Failure | 9 | .0043 | 18 | .0075 | 34 | .013 |
| Mechanisms | | | | | | |

Table 6.3. Range of Super Component Failure Rates for Turbine/Pump Test (λ in units of $E-6/hr$, q_d in units of per demand)

tem. Actual failure data taken from the operational history of the plant indicate that it has failed seven times during the life of the plant. As six of the failures resulted from binding of the valve seat, it is estimated that perhaps one half of the failures could have been caused by closure of the valve with too much force after a previous demand. This is supported by the fact that the valve failed to open after only a short standby period on Nov 11, 1974. If it assumed that the probability of failure is divided equally between standby and demand related failure mechanisms, a point estimate of the valve's failure rate yields:

$$q_d = \frac{7}{2(130)} \quad (6.1)$$

and,

$$q_\lambda = 1 - e^{-720\lambda} = \frac{7}{2(130)} \quad (6.2)$$

| Confidence Level | λ ($\times 10^{-6}$ /hr) |
|------------------|-----------------------------------|
| 95% | 57.0 |
| 90% | 43.8 |
| 80% | 30.6 |
| 75% | 26.4 |
| 50% | 13.2 |

Table 6.4. Upper bound estimates of MO 2301-3 Standby Failure Rate Since 1975.

The solution of the above equations then yield failure rates for MO 2301-3 of $\lambda = 3.8E-5/\text{hr}$ and $q_d = 0.027/\text{d}$.

Further inspection of the data indicate, however, that an engineering assessment of the cause of the failures was accomplished, and the last time the binding failure was observed was June 8, 1975). With this information, it may be valid to assume that the six binding failures were burn-in failures which should not be given full weight in estimating the failure rate of the valve. Given that no failures of MO 2301-3 have occurred during periodic tests since June 1975, one can calculate a one sided confidence interval for the valve failure rate using standard chi square approaches. Table 6.3 gives the failure rates at various confidence levels.

Finally, generic data is available from the Nuclear Plant Reliability Data System [NUREG-2232] for motor operated gates valves of size similar to MO 2301-3. Unfortunately the data is not grouped by standby environmental conditions, so the applicability of it is subject to question. Also, criticisms regarding the variances among the utilities in reporting failures have been raised. [EPRI1064]. Even though NPEDS may have large uncertainties, Never-the-less it is reasonable to at least consider those failure rates when making a judgement about the current failure rate of MO 2301-3. Generic failure rates for motor operated valves are given in Table 6.2.

Considering all the information available, the standby failure rate of MO 2301-3 is estimated to be between 5 and 25 E-6 per hour. The information on demand failure mechanisms is used to divide the overall super component failure rate. The lower limit takes into account the generic data on valves, but lies in the upper part of the range. The upper limit corresponds to the 75% confidence limit of no observed failures since 1975. That value is considered reasonable since the engineering problems with the valve appear to have been solved.

It should be noted that the valve has a monitored failure rate. The one additional failure of the valve (under-voltage coil, March 23, 1977) which occurred since 1975 was detected during the standby period by normal operator surveillance. (A periodic test was not necessary to detect it.) Since monitored failures occur randomly and are repaired when they occur, their contribution can be modeled on a per demand basis.

Unavailability During Testing

The second area of uncertainty in analyzing the turbine/pump operational tests is the effective downtime created by the test. It was shown in Chapter 2 that test caused downtime can come from three primary causes: 1) $q_o \tau$, unavailability to override the test mode, 2) $P_f(\tau + T_R)$, Probability of Test Caused Failures, and 3) $q_d T_R$, Repair of Demand Failures.

In the case of the HPCI System, only the second and third terms contribute to the effective downtime per test (EDT). The reasoning is as follows. Although an operational test requires that the pump discharge be routed through a full flow test return line, during the flow test the system is in closer alignment with its operating condition than when it is on standby. At the start of the test all but one of the active components transfer into their operating configuration. (MO 2301-8, the Injection Valve, remains closed.) The only misaligned components are the test return valves, and if either one closes and MO 2301-8 opens, the injection function is successfully initiated. During standby, at least eight different components must startup or change position. Therefore, it is reasonable to conclude that the HPCI System is in its most available state when it is undergoing a turbine/pump operability test.

The major disadvantage of performing a turbine/pump operability test is the potential for generating failures which would not inhibit the functioning of the system during an emergency, but during normal operations require the system to be made inoperable for repairs. This includes not only leaks and burst seals in critical components, but also failure of equipment whose purpose it is to keep the working environment in the vicinity of the system free of contamination. Eleven failure reports have been identified as resulting from this type of failure during approximately 120

tests. If the probability of test caused failures is assumed to follow a binomial distribution, where P_f is a constant failure probability per test, the 90% confidence interval is $0.05 < P_f < 0.13$.

Actual downtime to the completion of repairs is not recorded on all failure reports. Based on partial information it is estimated that the system is inoperable for an average of about 20 hours following a test caused failure. The product of repair time and P_f then yields an EDT in the range of 1 to 4 hours per test.

The third term, $q_d T_R$, contributes only a small portion of the effective downtime per test. The maximum estimated value of test observable q_d is $0.0075/d$, which is a small percentage of the contribution of P_f .

Results

Figure 6.2 is a mapping of the optimum test interval of the super component which represents the single component cut sets that are tested by the turbine/pump operational test. This figure was generated by repeated calculation of the OPTEST subroutine option which is included in FRANTIC II-MIT. The figure gives the optimum test interval for the system given any assumed standby failure rate between 1 and $50 \text{ E-}6$ per hour and an estimated downtime per test of one to six hours. The range of failure rates estimated for the HPCI turbine/pump single component cut sets is given by the horizontal bar. It is placed on the current test interval of 30

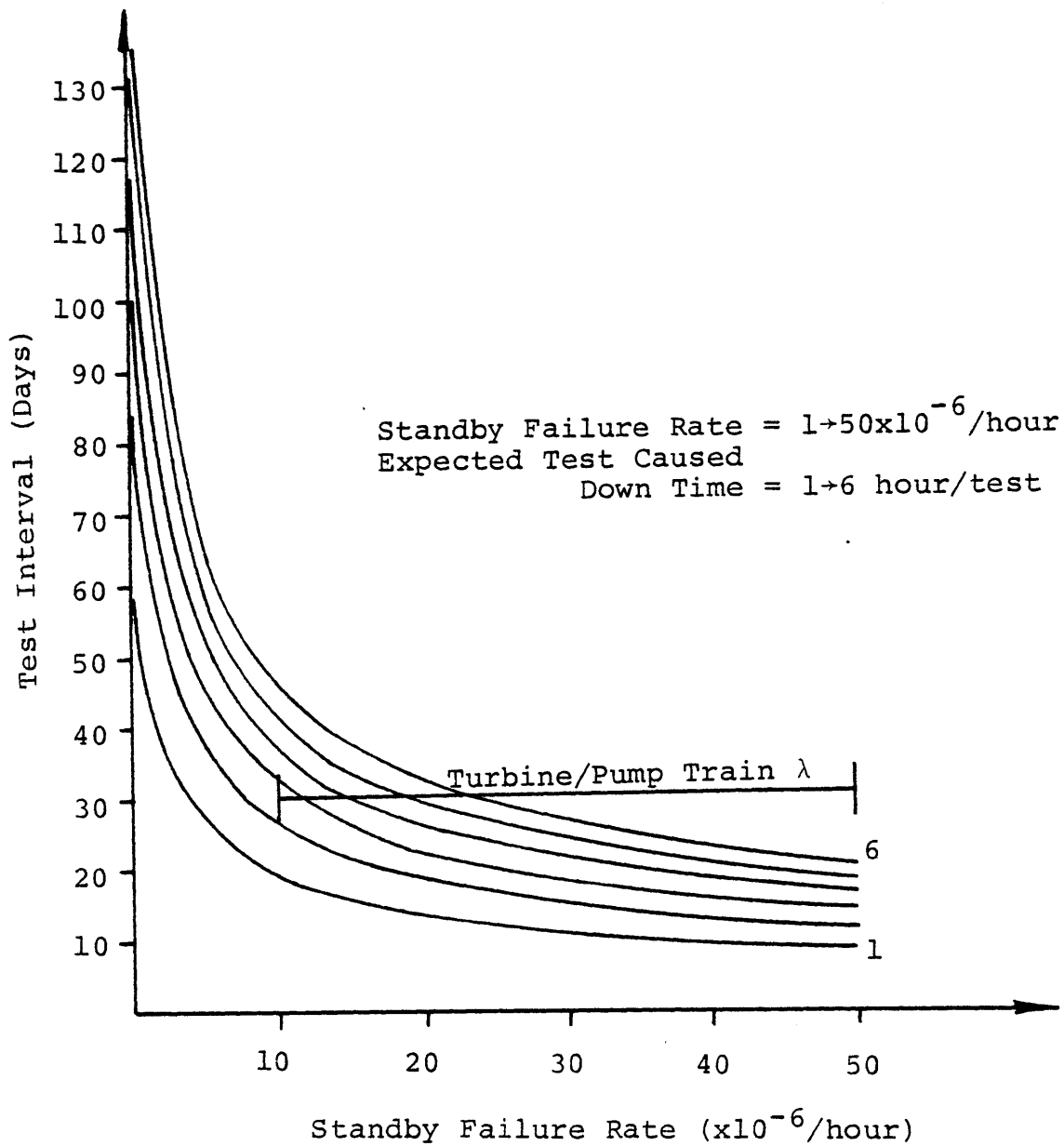


Figure 6.2. Optimum Test Interval of HPCI Turbine/Pump as a Function of Composite Failure Rate and Effective Downtime Per Test.

days. Note that for the lower half of the range of standby failure rates the 30 day test interval is very close to the optimum.

Figures 6.3 to 6.6 give an expanded version of the calculation for effective downtimes of one to four hours. These figures not only plot the optimum test interval, but also the contours for various factors of increase in unavailability due to standby failures. It can be seen from these figures that near the $q_{\lambda\min}$ contour line the average unavailability relative to its minimum value becomes insensitive to changes in both the failure rate and the test interval. Except for the case where the Estimated Downtime is one hour and the standby failure is above about $28E-6/hr$, the 30 day test is within 50% of the minimum attainable unavailability.

If demand failures form an appreciable percentage of the observed failures, the factor increase in the total unavailability of the system will be less than that shown in the figures, since

$$q_{av} = q_d + (1-q_d)q_{\lambda av} \quad (6.3)$$

Variations in the test interval do not actually produce the percentage change shown in Figures 6.3 to 6.6 in the overall unavailability of the system. Figure 6.7 illustrates this. It shows the results of OPTTEST calculations using the various ratios of demand and standby failure rates given in

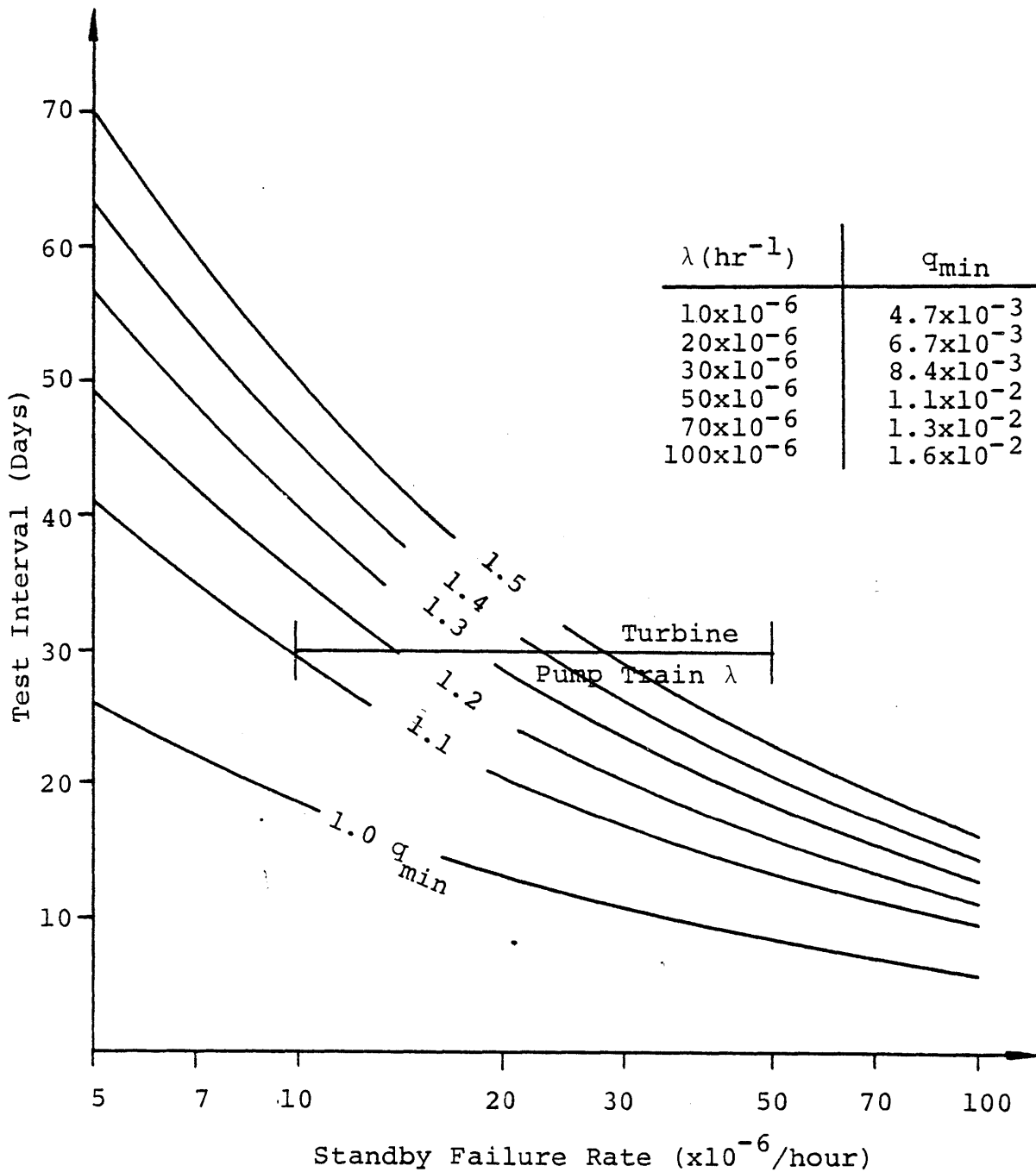


Figure 6.3. Average Unavailability Contours for HPCI Turbine/Pump Testing When Effective Downtime per Test is One Hour.

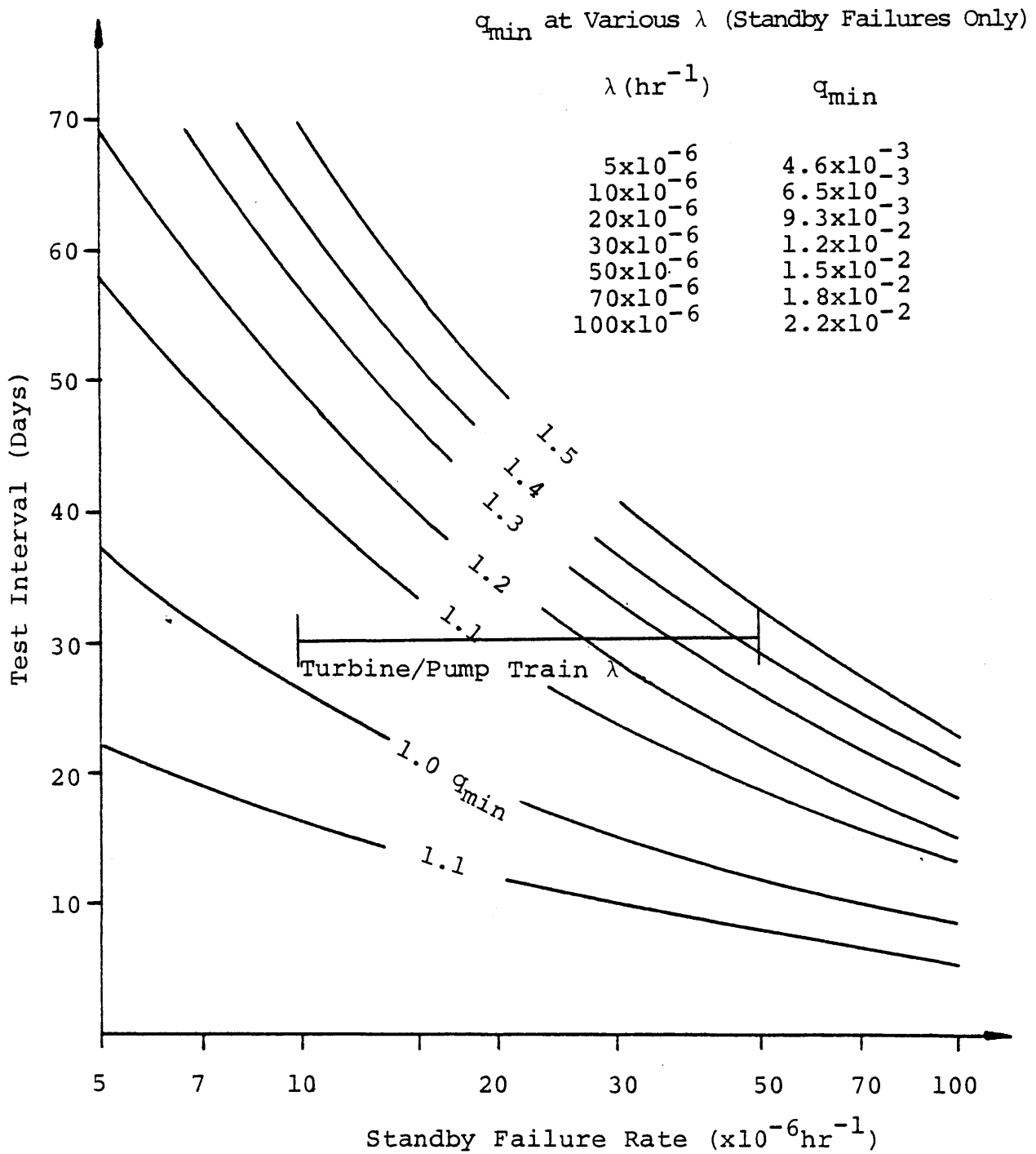


Figure 6.4. Average Unavailability Contours for HPCI Turbine/Pump Testing When Effective Downtime Per Test is Two Hours.

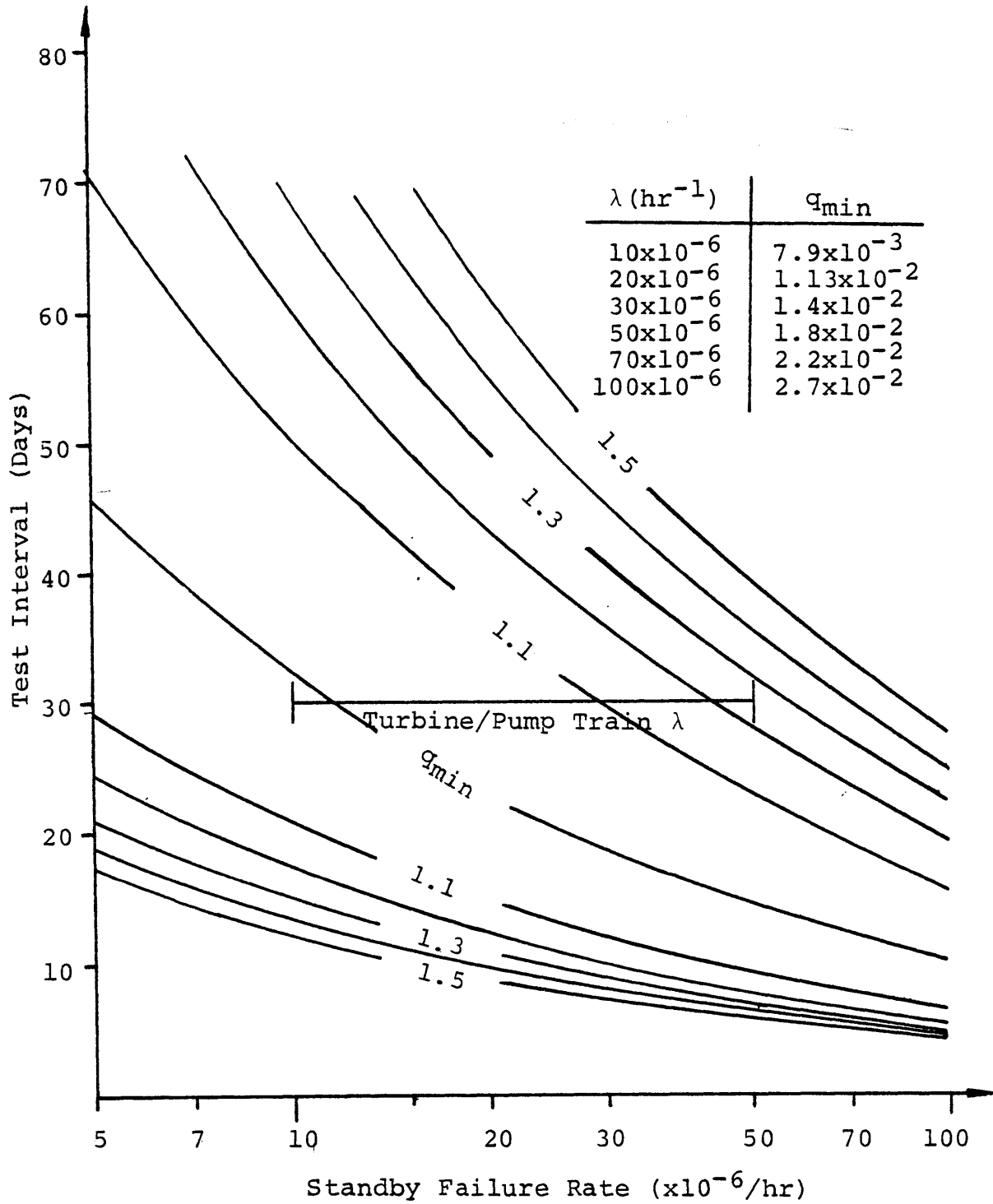


Figure 6.5. Average Unavailability Contours for HPCI Turbine/Pump Testing When Effective Downtime per Test is Three Hours.

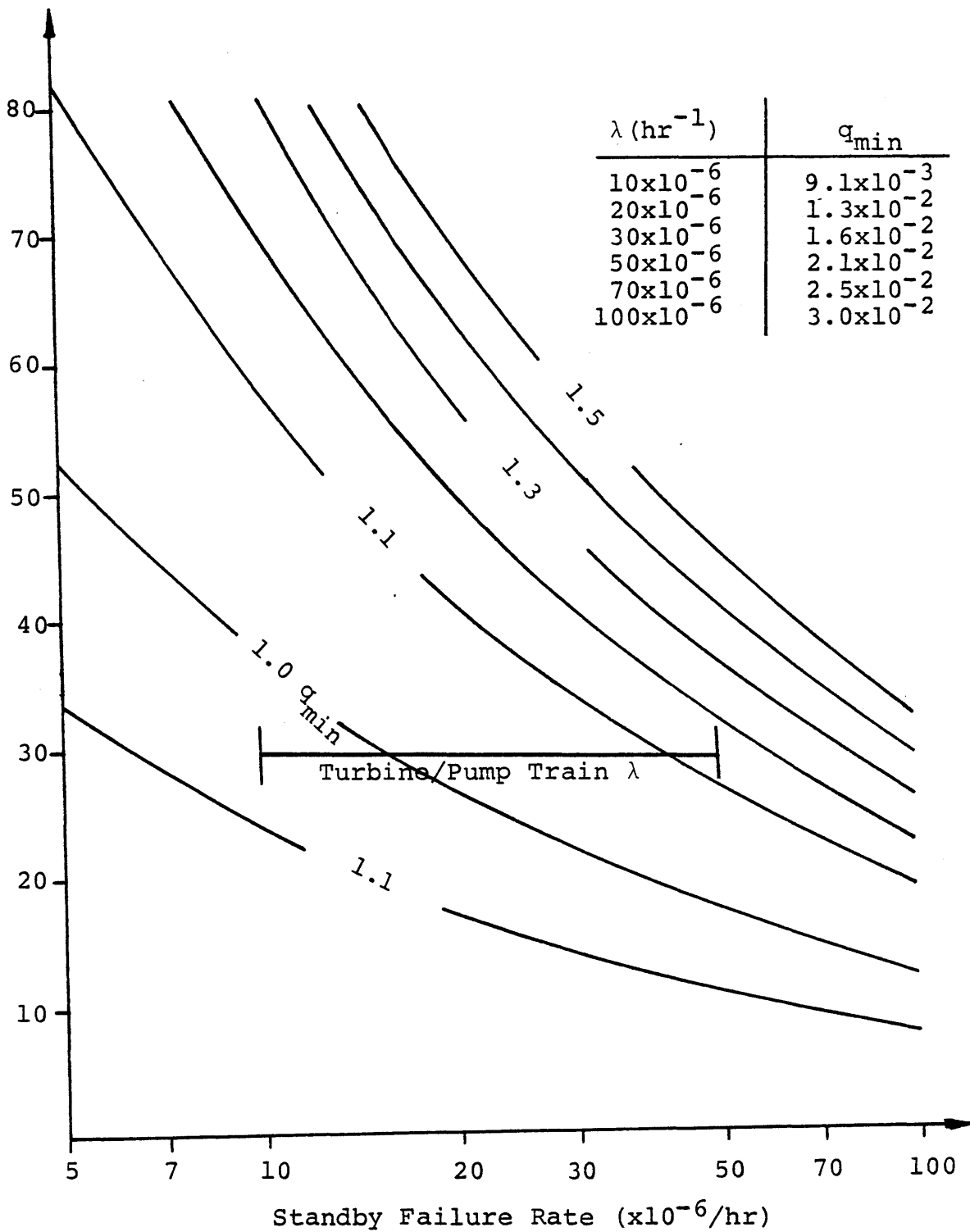


Figure 6.6. Average Unavailability Contours for HPCI Turbine/Pump Testing When Effective Downtime per Test is Four Hours

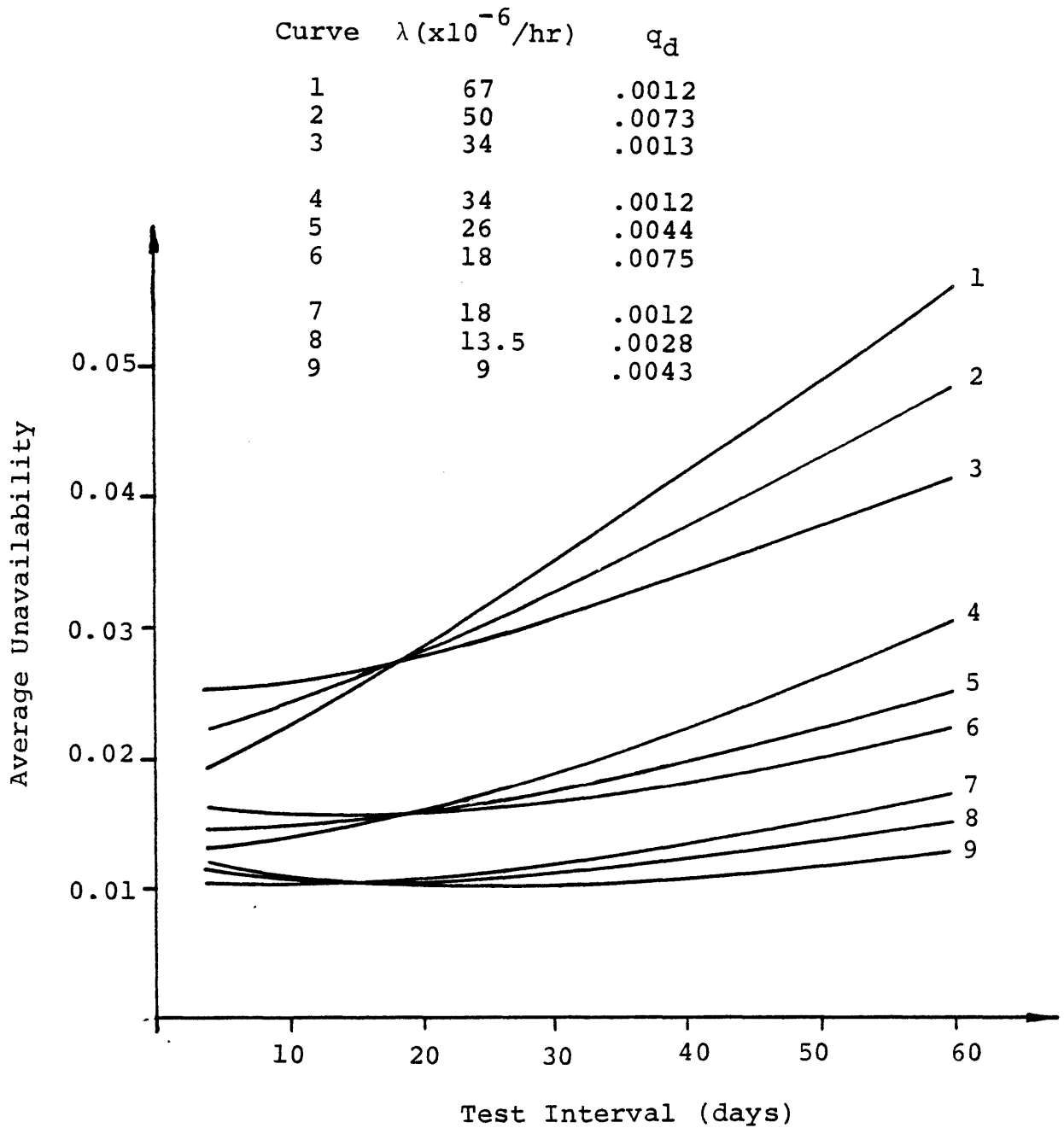


Figure 6.7. Super Component Unavailability Verses Test Interval for the Various Combinations of Demand and Standby Failure Rates Given in Table 6.3.

Table 6.3. Note that with the demand failure rates included in the failure data, the unavailability becomes less sensitive to the test interval.

Figure 6.8 shows the effect of test caused wearout on the on the average unavailability of the HPCI turbine/pump train at various times in the system's operational life as a function of test interval throughout the previous operational life. For these curves only standby failures are considered. The base curve is that obtained with no test caused wearout. For the other curves, it is assumed that each test causes an increase of 1% in the susceptibility of the components in the train. This is modeled by making the super component failure parameter $f_{\lambda} = 1.01$. The successive five year time frames are obtained by incrementing the Offset Time parameter for the super component by five year increments. This parameter is available only in FRANTIC II-MIT. The calculation shows that the existence of test caused wearout in a system favors accomplishing testing at longer intervals. Initially the shorter intervals reveal undetected failures sooner and thus produce lower unavailability. However, each test also increases the rate at which failures occur in the future. Consequently, the shorter test intervals produce more failures and a resultant higher average unavailability as the plant becomes older.

The failure data for the HPCI system does not appear to indicate that wearout is occurring in the turbine/pump com-

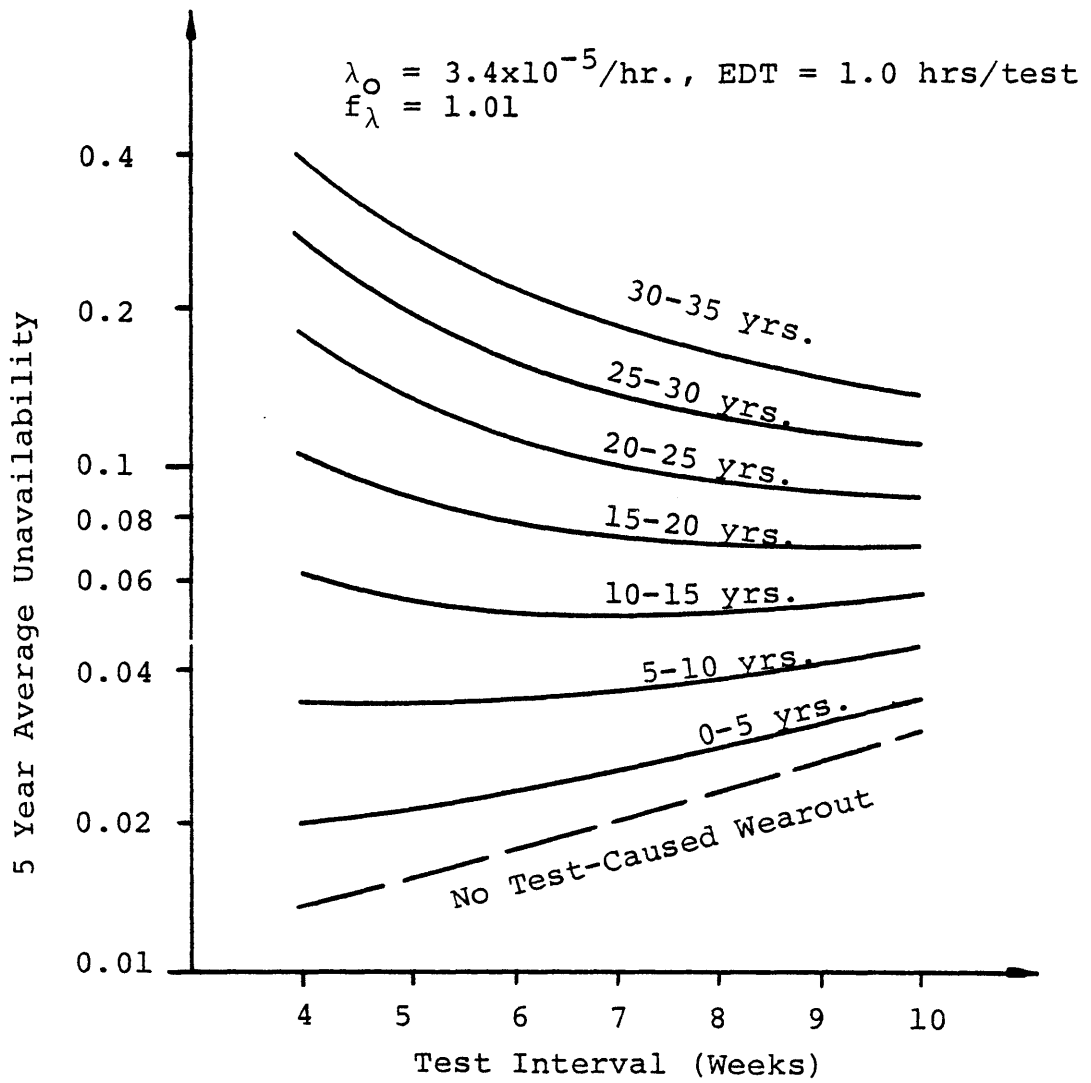


Figure 6.8. Effect of Test-Caused Wearout on Optimum Test Interval.

ponents. However, if inspection of the equipment reveals possible wearout, Figure 6.8 supports a lengthening of the operational test interval by up to several weeks. This will increase the average unavailability by only a few percent, but would increase the useful life of the component subject to test caused wearout.

Recommendations

The current test interval of 30 days appears to be well within the range that provides close to the minimum unavailability attainable for the system. No significant improvement in availability can be achieved by shortening the interval. The potential for additional test caused failures and increased test caused wear-out argues against shortening the test interval to obtain a marginal reduction in the time over which standby failures can remain undetected. Aside from the recommended consolidations, no change in the turbine/pump operability tests appears to be warranted.

6.5 AUTOMATIC INITIATION FUNCTION TESTS

Simplified diagrams of the HPCI automatic initiation function are given in Figures 6.9 and 6.10. The automatic initiation function requires that 1) either the low-low reactor water switches or the high dry well pressure switches activate their associated logic relays, and 2) the logic relays activate the appropriate circuits in the HPCI's active components. Failure to accomplish automatic initi-

Initiation Function Logic

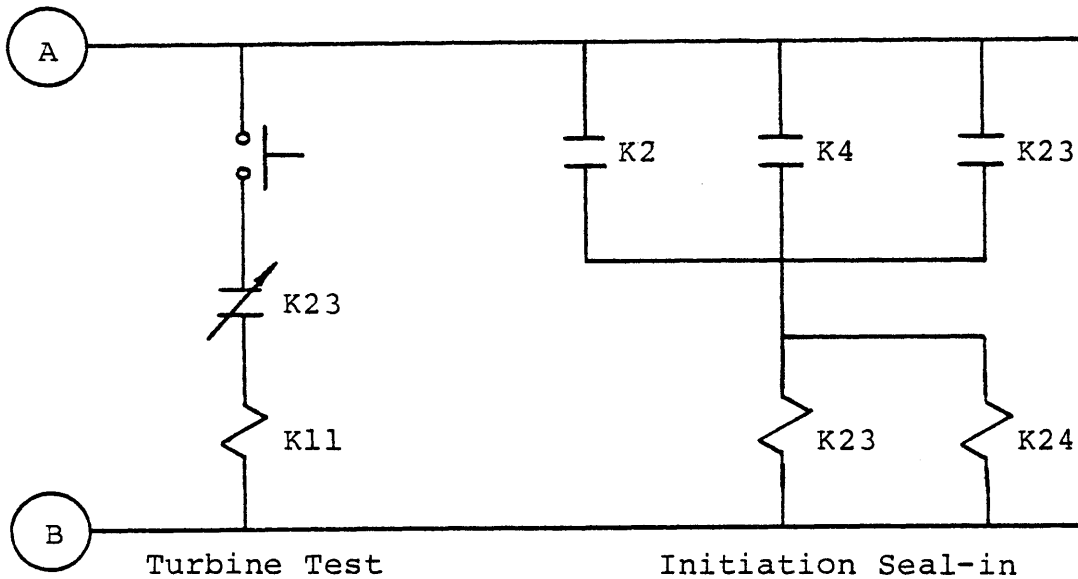
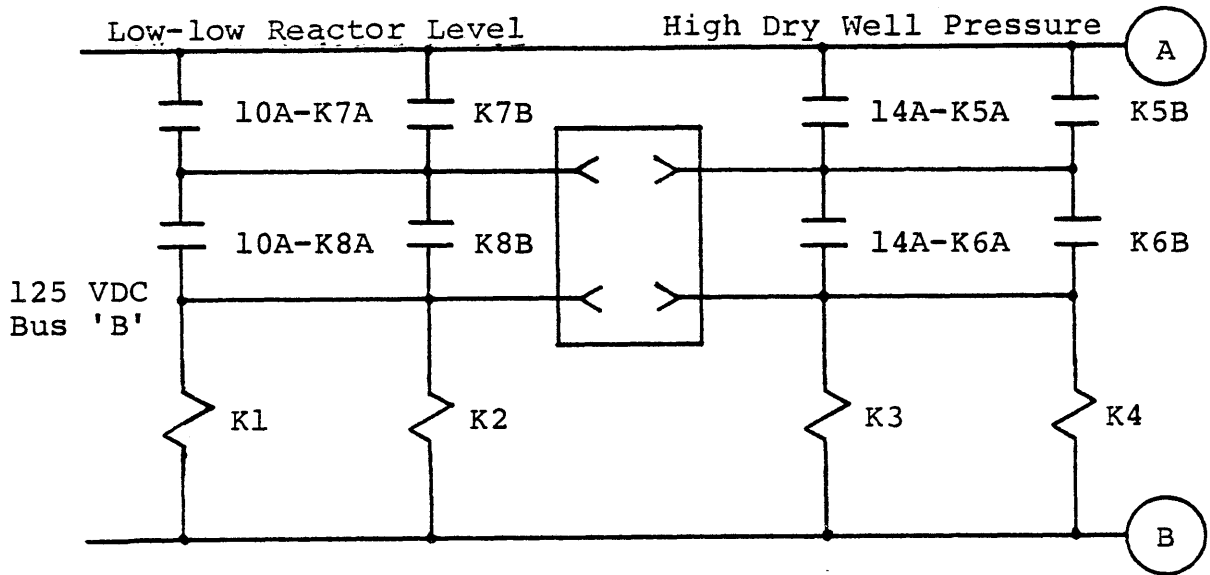


Figure 6.9. Simplified Diagram of HPCI Initiation Function Components. (Relay Signal Flow is given in Figure 6.10.)

| Component | Standby Condition | Signal To | Signal From |
|---|-------------------|-----------|-------------|
| *Steam Supply Valve MO 2301-3 | Closed | Open | K1,K3 |
| *Injection Valve MO 2301-8 | Closed | Open | K1,K3 |
| *Auxillary Oil Pump | Off | On | K24 |
| Minimum Recirculation Valve MO 2301-14 | Closed | Open | K2,K4 |
| CST Supply Valve MO 2301-6 | Open | Open | K1,K3 |
| Test Return Valve MO 2301-10 | Closed | Close | K1,K3 |
| Test Return Valve MO 2301-15 | Closed | Close | K1,K4 |
| Injection Valve MO 2301-9 | Open | Open | K2,K4 |
| *Seal-in Relay 23A-K23 | Open | Close | K2,K23,K4 |
| *Seal-in Relay 23A-K24 | Open | Close | K2,K23,K4 |
| Gland Seal Condensor | Off | On | K24 |
| Turbine Test Override | Open | Open | K23 |
| *Steam Isolation Valve MO 2301-4 | Open | Open | K2,K3 |
| *Steam Isolation Valve MO 2301-5 | Open | Open | K1,K3 |
| Seal-in Indicator on Operator Panel | Off | Lit | K24 |

Figure 6.10. Initiation Logic Signal Flow. Starred components are those currently verified to receive an initiation signal.

ation is assumed to produce system failure. Currently, five tests are accomplished on the initiation logic. Two tests verify the functioning of the two types of initiation sensors monthly and require a quarterly calibration. Three tests check the initiation logic relays semi-annually.

To assess the periodic testing policy for the initiation function, the intermediate event, "Failure to Generate Automatic Initiation Signal at Active Components" of the Injection Function Fault Tree is made into the Top Event of an intermediate level fault tree. (See Sheets 3-8, Appendix A.) Cut sets which contribute to this Top Event will also contribute to the more general failure definition of the HPCI System.

The evaluation presented in this section indicates that the initiation function is a relatively minor contributor to the overall system unavailability. This is expected, since the dominant contributors to overall system unavailability involve active components with relatively high failure rates. However, the analysis does reveal two areas where major improvements in the design and testing of the initiation function can be made:

- 1) The transmission of an initiation signal to 7 of the 11 components which receive it is currently not verified. Changes to the initiation logic tests can make this verification without necessarily requiring that each component

activate. Suggestions on how to do this are given in the section on the logic tests.

2) The analysis reveals three minor changes to the initiation circuit that eliminate the single failure contribution of Basic Failure Event 12 from the initiation function fault tree. These changes result in a decrease of over an order of magnitude in the unavailability of the function and simultaneously reduce the requirement for periodic testing of all the automatic initiation circuit components. Due to the lower unavailability created by the design changes, testing requirements on the initiation logic relays can be reduced to once per cycle.

6.5.1 INITIATION SENSOR TESTS

The following tests are currently accomplished to verify the operability of the initiation sensors:

Procedure 8.M.2-2.1.1. Reactor Water Level Safeguards System - Frequency: functional monthly, functional and calibration quarterly. This procedure tests the four reactor low-low level switches, which are wired in an one-out-of-two, taken twice configuration, as shown in Figure 6.9. During the test the logic remains active. A prerequisite for the test is that all four sensor switches are open. This verifies that none have shorted and reduces the chance of spurious initiation of the HPCI injection function during the test. In turn, each one of the switches is isolated and attached to a pneumatic calibrator. Pres-

sure to the inlet of the pressure sensor simulates differences between reactor water level and a reference water level and closure of the sensor's relay contact is verified. During calibration each switch is attached to a cold water head simulating the difference between reactor water level and reference level, and the actual difference in water head is measured.

The test isolates each one of the four switches for about 15 minutes. During most of that time, the switch is not activated. Because of the nature of the test, it is estimated that switch function can not be restored during the test time if a LOCA were to occur, yielding a q_0 of 1.0. The effect is to reduce the redundancy during the test to one-out-of-one for the switch wired in parallel to the one undergoing testing.

Procedure 8.M.2-2.1.4. Reactor Drywell High Pressure

- Frequency: functional monthly, functional and calibration quarterly. This procedure tests the four drywell pressure sensors, which are connected with the same logic as the low-low reactor water level pressure sensors. It follows the same procedure as 8.M.2-2.1.1 using a test signal appropriate for these sensors. Observations and comments for that procedure also apply to this test.

Quantitative Analysis

The sensitivity of initiation function average una-

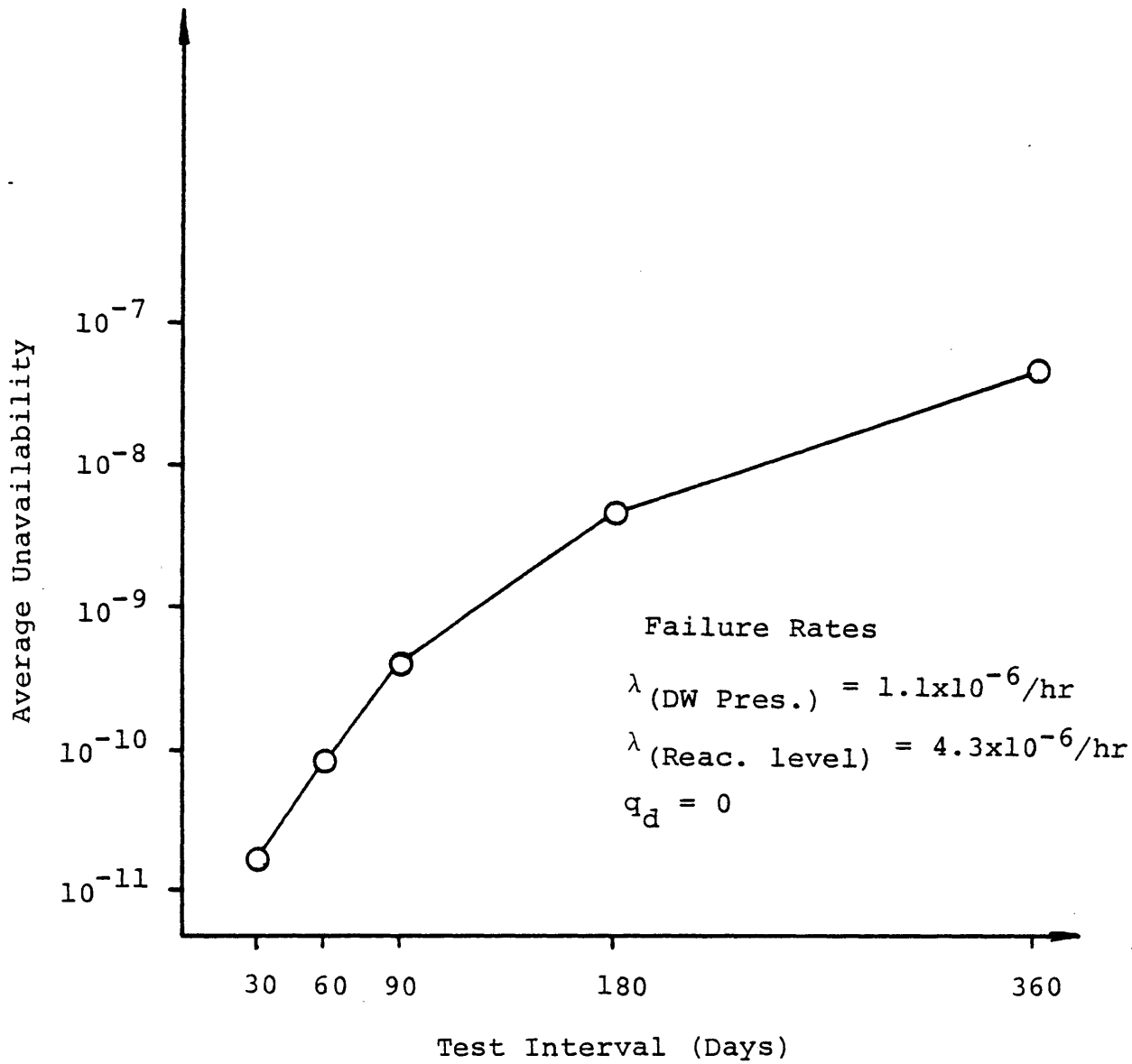


Figure 6.11. Average Unavailability of Initiation Sensors Without Common Cause Failures.

vailability to variations in the test interval of the temperature and pressure sensors, is shown in Figure 6.11. For this analysis it is assumed that either the low-low water reactor water level or high dry well pressure sensors are capable of detecting a small LOCA. Therefore, the sensors are both diverse and redundant and would be expected to produce a relatively low unavailability. Figure 6.11 shows that the probability of having neither sensor group available to detect a LOCA is extremely small, given 1) there are no common cause failures within a given group and 2) logic relays are operational.

When common cause failures are considered, the unavailability of a particular group of sensors is given a single failure event which relates the probability that all the sensors of that group fail simultaneously. Figure 6.12 illustrates the potential effect of these common cause failures, which are accounted for by Events 6 and 7 in the HPCI fault tree. For this analysis it is assumed that the two diverse means of detecting a small LOCA makes the possibility of a common cause failure taking both types of sensors off line remote.

For the purpose of estimating the possible effects of common cause failures, standby failure rates corresponding to 10% and 100% dependent failures and demand failure rates of 0.0 and 0.001 are considered. The demand failure rates account for the possibility that the conditions of the true

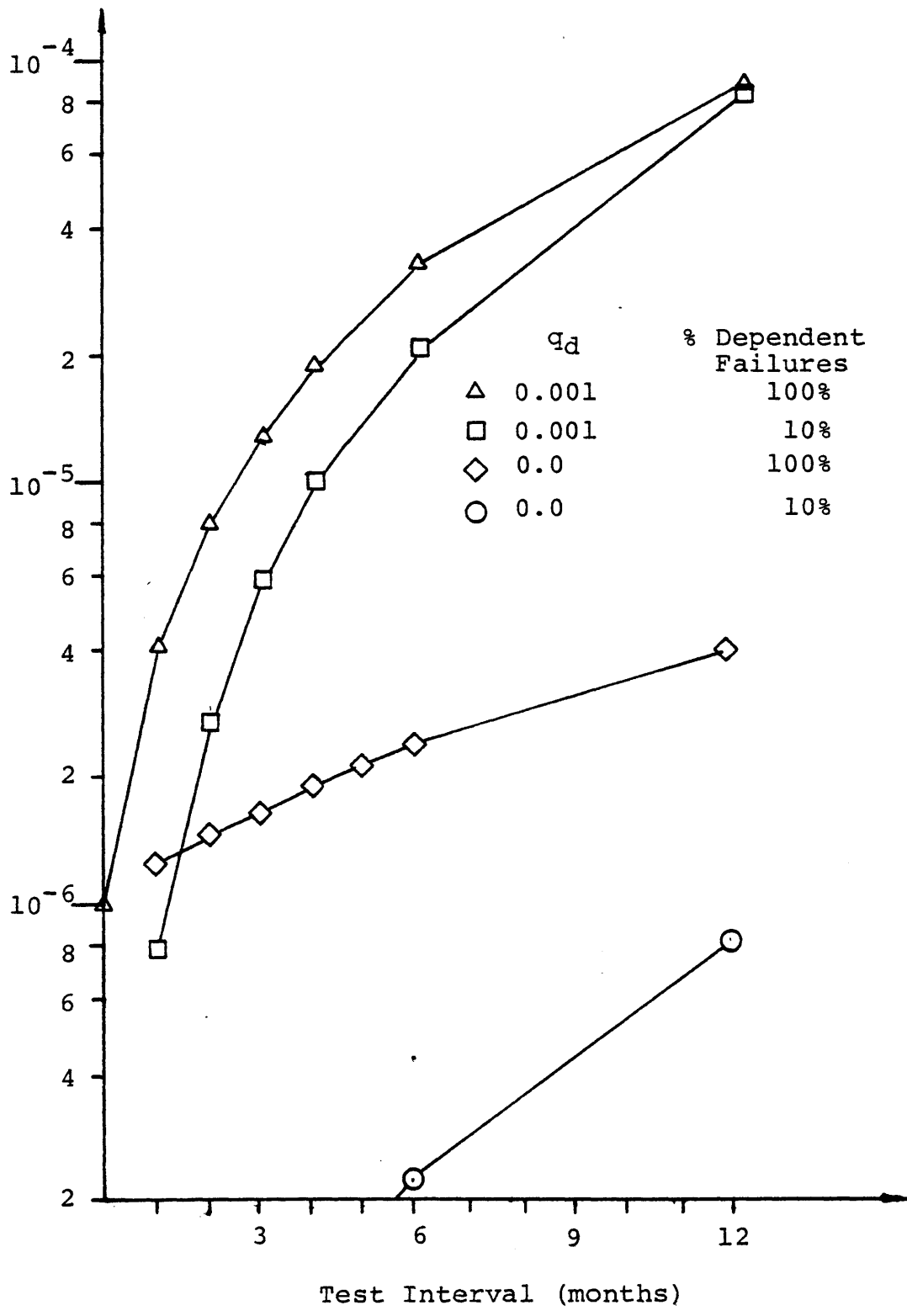


Figure 6.12. Average Unavailability of Initiation Sensors With Common Cause Failures Within Individual Groups of Sensors.

demand prevent the sensors from responding in accordance with design. For the purposes of this sensitivity study, it is estimated that this probability is not more than 0.001. The "worst case" standby failure rate assumes that sensors failures are completely dependent, so common cause failures occur at a rate equal to the failure rate of an individual component. Somewhat arbitrarily, a second common cause failure rate equal to 0.1 of the component failure rate is also used for the estimate. (The very low average unavailabilities obtained for the the sensors, even assuming completely dependent failures, indicate that more detailed analysis is not warranted.) It can be seen in Figure 6.12 that it requires a large contribution of common cause failures to make the average unavailability of the sensors go above $1.0E-5$. That does not occur until the standby failure rate is equal to the individual component failure rate and the demand failure rate is close to 0.001.

Despite the very low unavailability obtained for this function, we do not recommend increasing the test interval. These two groups of sensors activate more than just the HPCI System. The level sensors also contribute to the activation of the Reactor Core Isolation Cooling System, Automatic Depressurization System, Standby Diesels, Low Pressure Coolant Injection System, and the Core Spray System. The pressure sensors also contribute to the activation of the Low Pressure Coolant Injection System and the Core Spray

System. For this reason it is reasonable to continue testing them at the current 30 day interval to insure that their unavailability remains very low.

6.5.2 DESCRIPTION OF INITIATION LOGIC TESTS

The initiation logic consists of relays associated with either the low-low reactor water level switches or the high drywell pressure switches, an initiation signal seal-in relay, and a relay to activate the controller. Figures 6.9 and 6.10 show the signal flow of these relays. When the low-low reactor level sensor contacts close in a 1-out-of-2, taken twice, logic relays 23A-K1 and K2 (designation abbreviated on Figure 6.16) are activated. Relay K2 closes a contact in a circuit which activates K23 AND K24, and the four relays energize the specific components shown in Figure 6.10. The high dry well pressure sensors accomplish the same function, but energize K3 and K4 instead of K1 and K2.

As the relay logic is currently designed, at least four relays must function for successful activation of the system. Relays 23A-K1 and 2 are redundant with 23A-K3 and 4, provided that both the low-low reactor level and the high drywell pressure sensors are capable of detecting the LOCA. However, essential functions are also initiated by two non-redundant relays, 23A-K23 and K24.

Current Test Policy

There are currently three procedures for testing the

initiation logic:

Procedure 8.M.2-2.10.4.2. HPCI Initiation Logic Test.

Procedure 8.M.2-2.10.4.3. HPCI Steam Supply Isolation Valve Logic.

Procedure 8.M.2-2.10.4.4. HPCI Injection Valve Logic.

Each one is accomplished in approximately the same manner. First circuit breakers to most active components are opened. Then the low-low reactor water and high drywell pressure switches are closed in a sequence which tests their wiring logic and the activation of the required logic relays is verified. The procedures differ primarily in the components which are kept active during the test. Figure 6.10 shows that 11 different components receive signals from the initiation logic. The tests verify that only four of these components receive and function in response to the signal, specifically:

- 1) 8.M.2-2.10.4.2 - Auxillary Oil Pump.
- 2) 8.M.2-2.10.4.3 - MO 2301-4 and MO 2301-5. (These valves are verified to open on an initiation signal, given they are closed. The only time when they will be closed during normal operation is during testing of the autoisolation signal function.)
- 3) 8.M.2-2.10.4.4 - MO 2301-8.

Comments on Current Test Procedures

Before proceeding with a quantitative determination of recommended test intervals, two comments are in order:

1) The three initiation logic tests should be consolidated into one procedure which verifies that all 11 components receive the necessary initiation signal. In most cases the signal path can be checked without requiring that the component itself activate. For example, in the circuit which opens MO 2301-3, the manual switch is parallel to the automatic initiation circuit contacts. Therefore, closure of the initiation contact should produce a short circuit across the manual switch. Then activation of the valve by the manual switch would by inference verify activation by the automatic initiation circuit.

2) Accomplishing the logic tests in conjunction with the initiation sensor tests will provide an integrated test of the entire logic train. If the logic tests are done during annual refueling, as recommended in the following sections, the longer time required for the integrated test will not contribute to the system's unavailability.

6.5.3 QUANTITATIVE EVALUATION OF INITIATION LOGIC TESTS

Failure Event 5 models system unavailability resulting from the initiation logic tests. It has been given a standby failure rate of $1.0E-26/\text{hr}$ to "switch on" the periodic test logic of the code. System downtime for injection logic testing is then modeled using q_0 , τ , and T_2 derived from analysis of the logic tests. Figure 6.13 shows the contribution of Failure Event 5 to average system unavailability. (Although the data points were generated by the FRANTIC

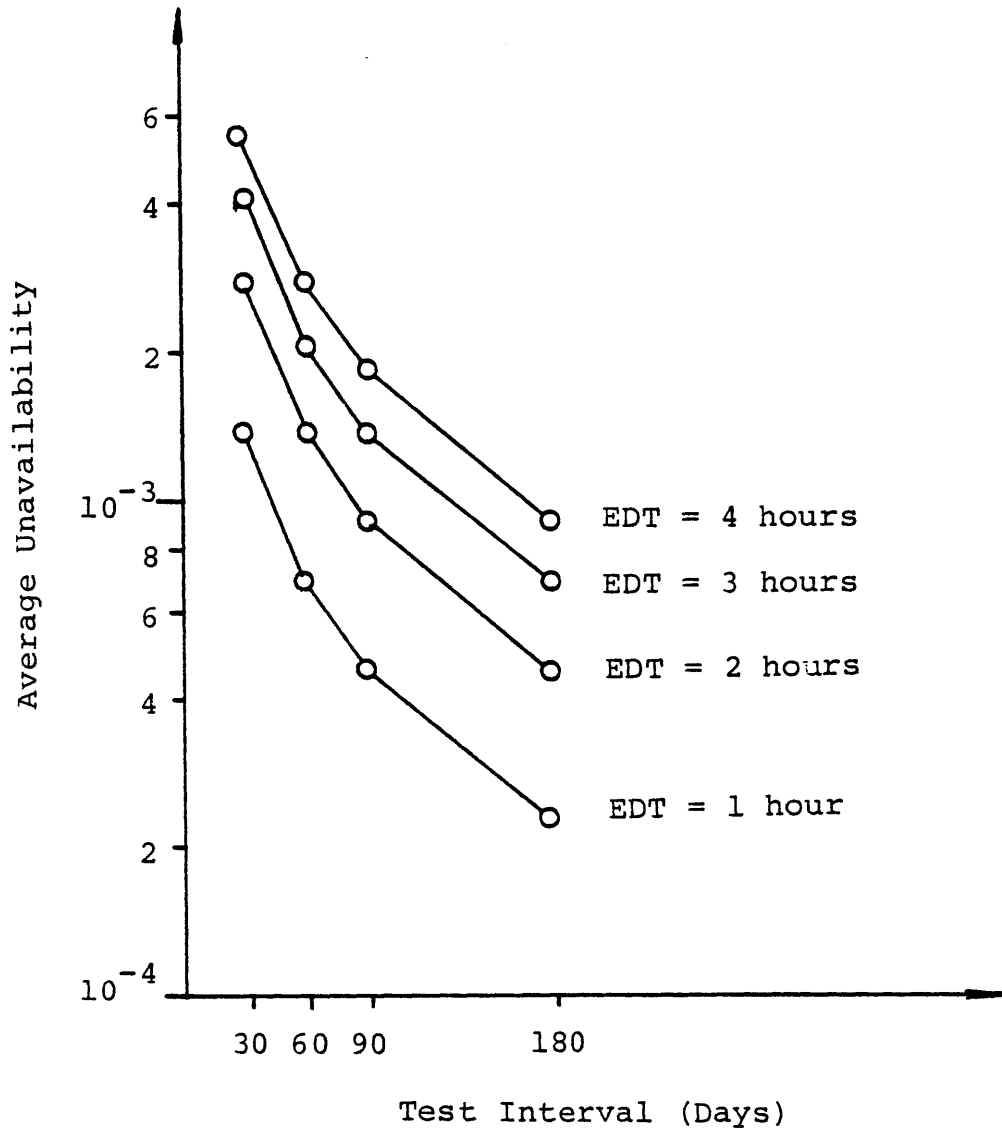


Figure 6.13. Average Unavailability Due to Online Testing of the Initiation Logic Relays.

II-MIT code, they could easily be calculated using the idea of effective downtime. The resulting average unavailability would be $q_{av} = (q_o \tau) / T_R$. An important consideration in the quantitative analysis is the fact that if the logic tests are done when the reactor is down for refueling and maintenance they do not contribute to the system's unavailability.

Unavailability Using Current Logic Design

To obtain a comparison with the current design and test policy, the first series of calculations assumes that three different tests of the HPCI initiation logic will continue to be made, but with procedures modified so that proper transmission of the initiation signal to one third of the active components will be verified by each test. (If the procedures are not changed, a failure in the circuit from the initiation relays to one of the seven unverified components will remain undetected until a true demand, and the probability that such a failure has occurred will increase monotonically throughout the life of the plant. Since the procedural change is reasonable to implement, the magnitude of the undetectable is not estimated.) With this policy, the automatic initiation of each active component will be tested every six months, which is the intent of the current policy.

The current logic tests result in the HPCI System being disabled for approximately one hour per test. During the

test circuit breakers to most of the active components are opened to prevent inadvertent injection into the reactor due to the test initiation signal, and it is conservatively estimated that there is a 0.5 probability that the system can not be activated in the event of an actual demand. This yields an Effective Downtime (EDT) of 0.5 hours per test. For the initial calculation it is assumed that an additional two or three components can be verified to activate without adding significantly to the EDT of an individual test.

Under the existing schedule the three logic tests are all accomplished during the same month. With this schedule, the second and third tests have little opportunity to detect standby failures in the relays, since there is little time for them to occur. Consequently, a policy similar to that currently being used would result in the relays being tested once every six months with a test time of 3.0 hours and an unavailability to override the test, $q_0 = 0.5$, yielding an EDT of 1.5 hours per logic relay test.

Figure 6.14 shows the effect of staggering the three logic tests. When the three tests are staggered, the number of tests accomplished in a six month period remains the same. However, since every test requires tripping the initiation relays, a staggering policy would result in their being tested at the staggered interval instead of once every six months. Also, since tests are no longer being made sequentially, the test duration for any given month

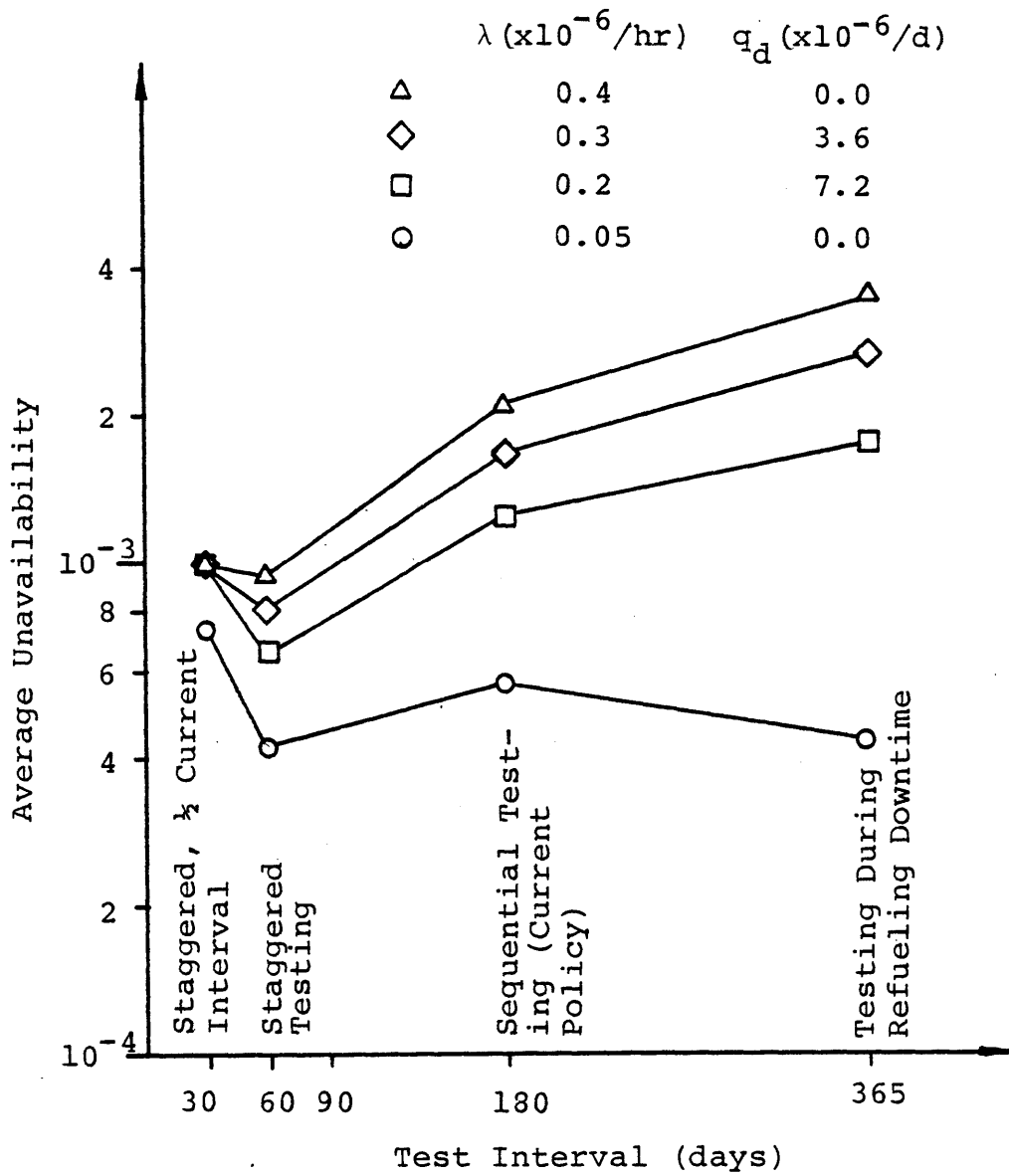


Figure 6.14. Unavailability of Initiation Logic Relays With Current Design and Current Test Procedures and Staggering.

decreases. The calculations are made for a variety of assumed failure rates which cover those expected for control relays. A 365 day calculation is also made. At this interval testing is done when the reactor is down, and EDT = 0. It can be seen in Figure 6.14 that with the current design the staggered testing policy yields the lowest unavailability. Note also that testing more often (each test every 3 months, with a resultant staggering interval of 30 days) increases the average unavailability, because of the unavailability to override the test.

Initiation Logic Relay Modifications

A series of three design improvements will now be presented, and their effect on the most reasonable testing policy of the initiation logic will now be discussed. For calculations showing the effects of the modifications, the following two assumptions are made:

1) The logic relay tests are consolidated into one procedure which verifies that each active component receives an initiation signal. It is estimated that the consolidated test will take longer than the current partial tests. For purposes of estimation the calculations are performed using an Effective Downtime (EDT) per test ranging from one to four hours.

2) The failure parameters $\lambda = 3.0E-7/\text{hr}$ and $q_d = 3.6E-6/\text{d}$ are reasonable representations of the relay failure rates. These data are derived from WASH 1400 assuming

25% demand failures and converting the reported demand failure rate to a standby failure rate using a 30 day periodic test interval.

The first recommended design change is to consolidate the functions of the two relays in the seal-in circuit. Currently, both relays 23A-K23 and 23A-K24 must function for initiation to succeed. However, Relay 23A-k23 has sufficient spare contacts to take up the functions of 23A-K24. The modification will remove relay 23A-K24 from Failure Event 12, a single component cut set in the injection function fault tree.

The second puts Relay 23A-K24 in parallel with Relay 23A-K23 across five existing sets of contacts in Panel 934. This modification is represented in the fault tree by making 23A-K24 Failure Event 96. Event 96 is then connected through an AND gate to Failure Event 12, making Event 12 part of a two component cut set. Figures 6.15 and 6.16 illustrate the changes to the logic circuits.

Modification 3 permits either the high dry well pressure sensors or the low-low reactor water sensors to activate both sets of initiation logic relays. The modification can be done by making the shunt from C to F on Test Connector 23A-J1A permanent. It reduces the number of cut sets in the initiation function fault tree from 79 to 49. As this modification reduces the separation of the two sensor groups,

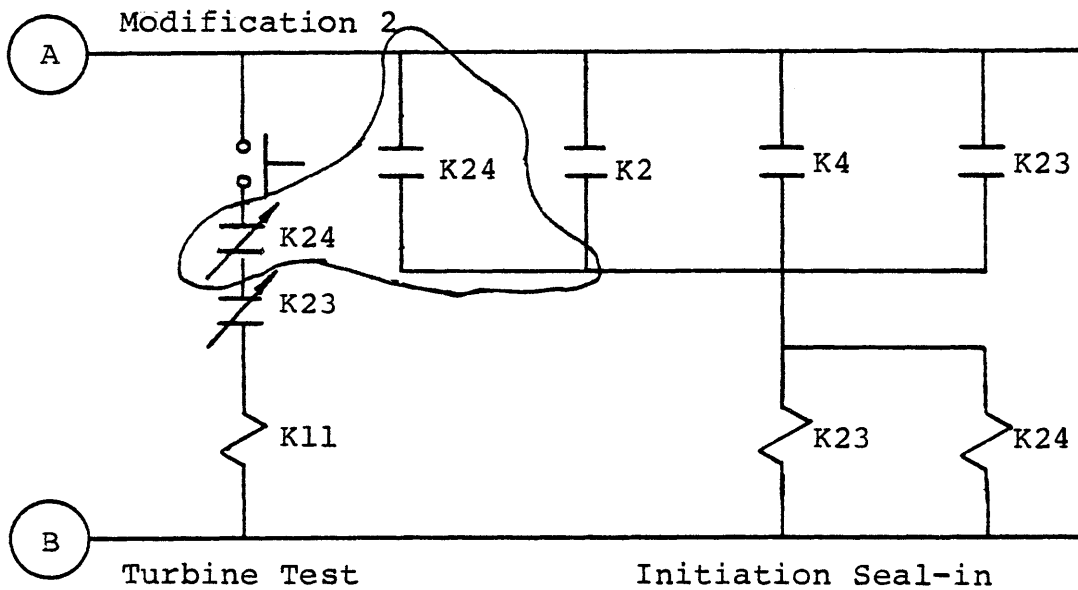
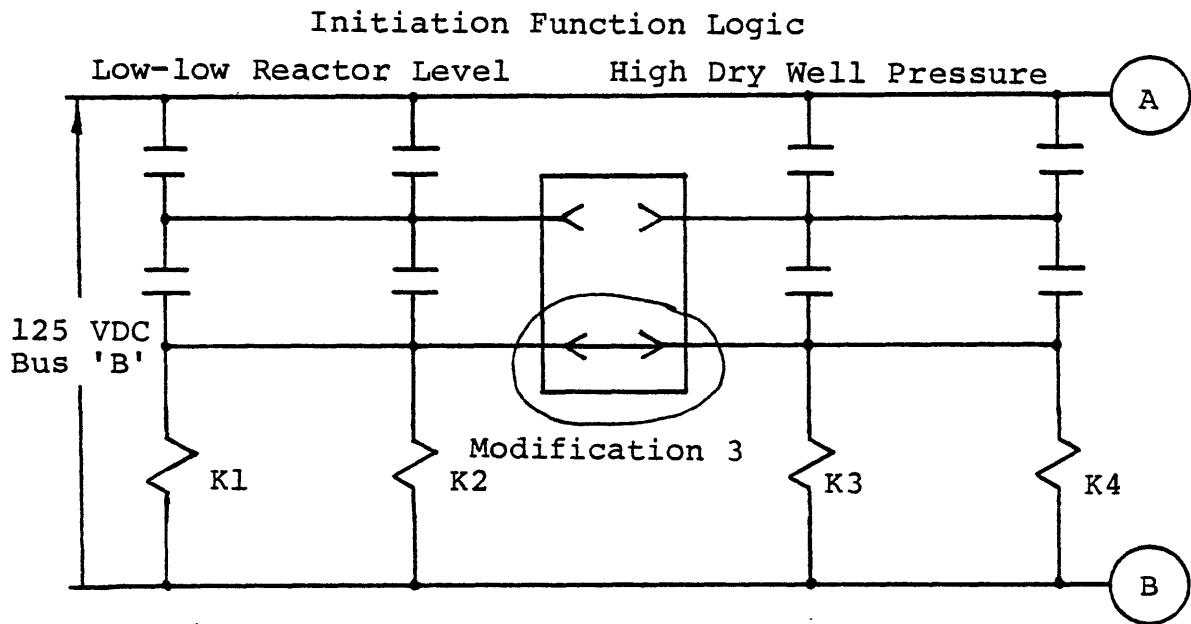


Figure 6.15. Wiring Diagram of HPCI Initiation Logic with Modifications Two and Three. (Additions to Relay Signal Flow are given in Figure 6.17.)

| Component | Standby Condition | Signal To | Signal From |
|---|-------------------|-----------|-------------------------|
| Steam Supply Valve MO 2301-3 | Closed | Open | K1, K3 |
| Injection Valve MO 2301-8 | Closed | Open | K1, K3 |
| Auxillary Oil Pump | Off | On | K24, <u>K23</u> |
| Minimum Recirculation Valve MO 2301-14 | Closed | Open | K2, K4 |
| CST Supply Valve MO 2301-6 | Open | Open | K1, K3 |
| Test Return Valve MO 2301-10 | Closed | Close | K1, K3 |
| Test Return Valve MO 2301-15 | Closed | Close | K1, K4 |
| Injection Valve MO 2301-9 | Open | Open | K2, K4 |
| Seal-in Relay 23A-K23 | Open | Close | K2, K23, K4, <u>K24</u> |
| Seal-in Relay 23A-K24 | Open | Close | K2, K23, K4, <u>K24</u> |
| Gland Seal Condensor | Off | On | K24, <u>K23</u> |
| Turbine Test Override | Open | Open | K23, <u>K24</u> |
| Steam Isolation Valve MO 2301-4 | Open | Open | K2, K3 |
| Steam Isolation Valve MO 2301-5 | Open | Open | K1, K3 |
| Seal-in Indicator on Operator Panel | Off | Lit | K24, <u>K23</u> |

Figure 6.16. Changes in Relay Logic Signal Flow Produced by Modification Two.

it should not be accomplished if sensor shorts to ground are a concern.

The modifications should not increase the potential for inadvertent initiation as three relays must still energize to activate all necessary components. It provides a more consistent design in that all three relays would now be redundant.

Summary and Recommendations

Figure 6.17 summarizes the effects of the three recommended modifications on the unavailability of the initiation function when the Effective Downtime per test is 2.0 hours. With the modifications there is a clear advantage to testing the logic relays only once every cycle. The unavailability for the 60, 90, and 180 day test intervals are almost entirely due to the unavailability to override the test. A once per operating cycle test policy eliminates Failure Event 5, which accounts for unavailability during online testing, from the initiation function fault tree. The unavailability for a 365 day test interval is due entirely to failures, since the test is done during scheduled downtime. Note that its unavailability is over an order of magnitude smaller than the contribution of the tests.

The current test procedures should be consolidated into one comprehensive procedure that verifies that all active components receive the initiation signal. In addition, minor design changes to the initiation logic relays

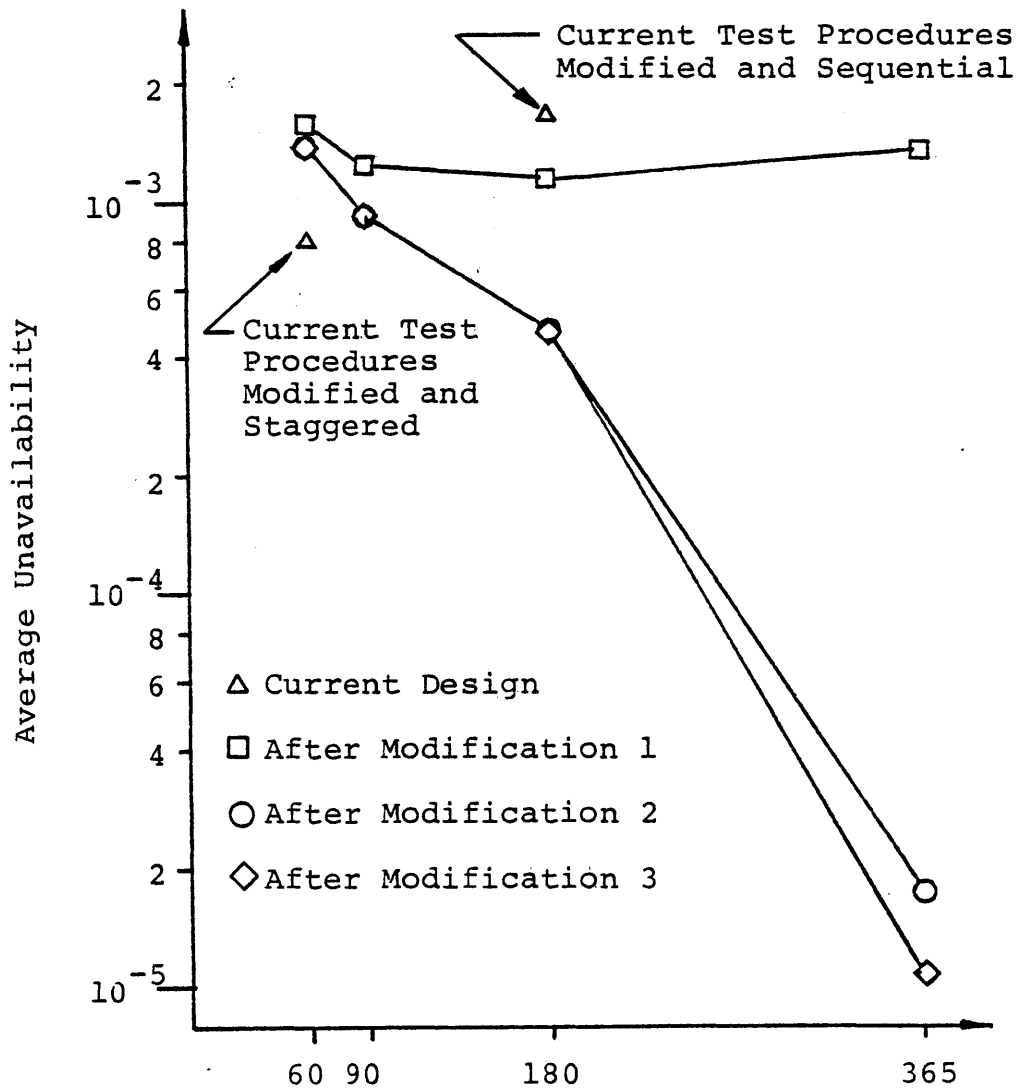


Figure 6.17. Effects of Design Modifications on Initiation Function Unavailability Verses Periodic Test Interval. Effective Downtime per Test = Two Hours.

will eliminate two single cut sets (Failure events 5 and 12) from the injection function fault tree. This reduces the requirement for logic tests from the current six per year to one per refueling cycle while producing a factor of 80 decrease in the unavailability of the initiation function.

To check the sensitivity of these recommendations to the failure rate of the relays, a calculation was made with the relay standby failure rate increased by a factor of ten to $3.0E-6$. Figure 6.18 shows the results of this calculation. Comparison of this figure with Figure 6.13 at 60, 90, or 180 day intervals shows that the contribution of the test downtime to the initiation function unavailability is significant compared to the contribution of standby failures even for this elevated failure rate. The unavailability for the once per cycle test jumps much higher as the failure rate increases. However, with the recommended modifications a once per cycle testing policy still produces the least average unavailability if the effective downtime for an online test exceeds two hours.

6.6 AUTOISOLATION FUNCTION TESTS

Currently two procedures are used to verify the steam line break sensor functions. A third procedure verifies the low pressure sensor functions. All three of these procedures test the entire train of the autoisolation logic and cause both Steam Line Isolation Valves, MO 2301-4 and 5, to

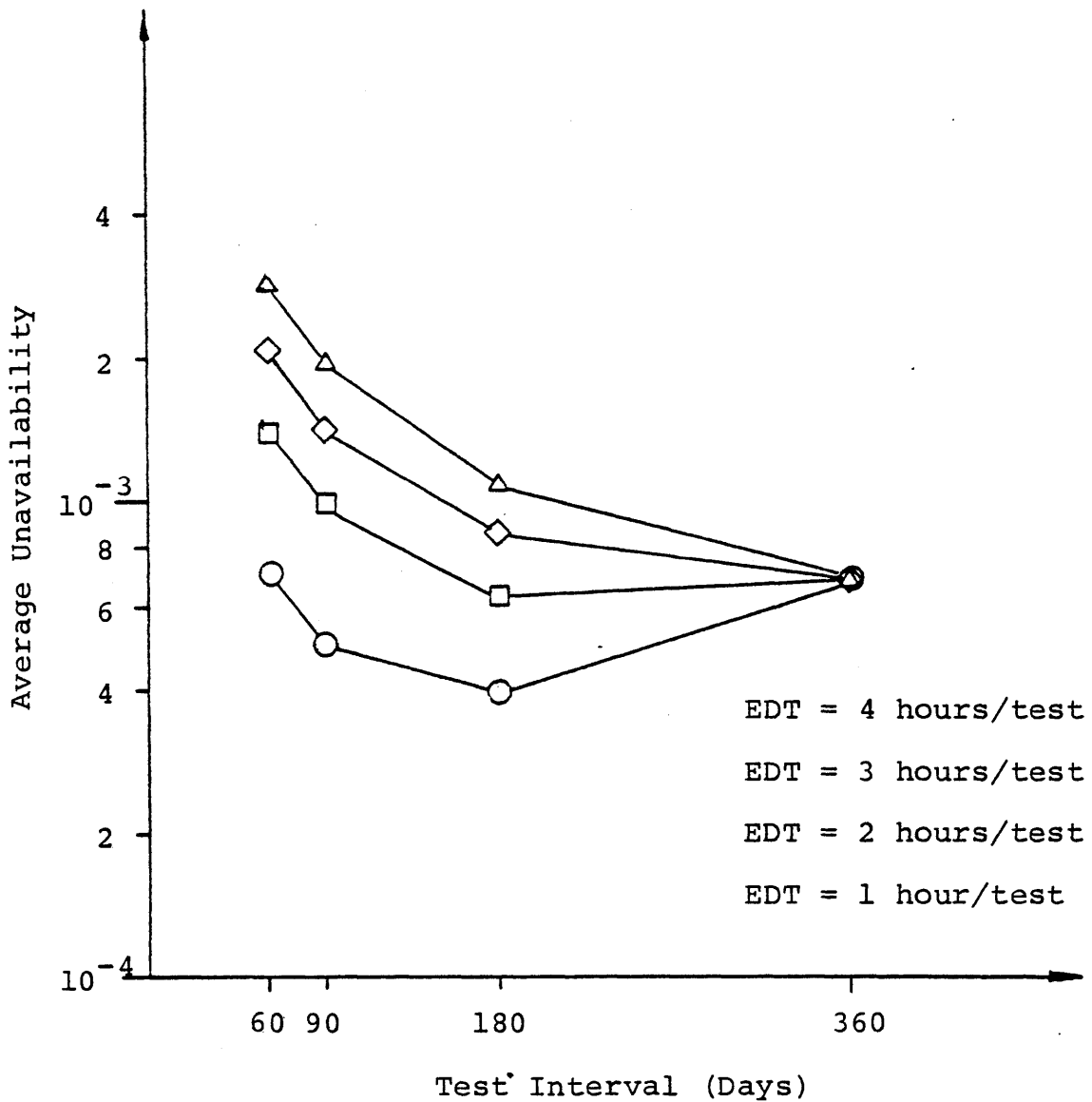


Figure 6.18. Sensitivity of Modified Initiation Logic Circuit Unavailability to an Order of Magnitude Increase in Component Failure Rates. With the recommended modifications, a once per operating cycle test interval still produces the least function unavailability.

close. A fourth procedure (8.M.2-2.10.5) tests the autoisolation logic semi-annually. This procedure is redundant with the three sensor tests and can be dropped without affecting the safety of the plant.

Before the tests are quantitatively analyzed, they will be described and some suggestions will be made for improving their ability to check the functioning of all isolation circuits.

Procedure 8.M.2-2.5.1. HPCI Steam Line High Flow

Isolation - The purpose of this test is to verify that the differential pressure switches in the steam supply send close signals to the steam supply line isolation valves upon sensing a high differential pressure. A sensor is disconnected from its tap line and a test differential pressure is applied using compressed air. (The ability of the tap line to transmit the pressure from the tap point on the steam supply line is not verified.) The first sensor check must produce closure of the Steam Line Isolation Valves, which are then kept closed for the remainder of the test. The sensor is then returned to service and the second is checked in a similar fashion. However, since the seal-in circuit is activated by the first test, only the closure of pressure switch contacts is verified by the remainder of the test procedure. The seal-in circuit keeps the logic relays closed, so the ability of the pressure switch to activate the autoisolation circuit can not be checked. (Note - This

problem is common to the next autoisolation functional test as well.)

Provided that it does not interfere with other operator duties, it is recommended that the operator stationed at the HPCI control panel during this test reset the autoisolation circuit before each instrument check so that the ability of each differential pressure switch to generate a close signal in the autoisolation relays can be checked. The isolation valves do not have to be reopened for this verification.

Procedure 8.M.2-2.5.3. HPCI Steam Line High Temperature - Two combinations of two-out-of-two temperature switch closures are checked in each of three different locations: 1) Torus Room, 2) Valve Station above 23 feet, and 3) Turbine/Pump Room. The test verifies a total of 6 circuits and 12 switches. Each circuit is checked by jumpering one switch and applying a high temperature to the series sensor. The jumper is then reversed, and the second sensor is tested. This procedure verifies both that the sensor has not inadvertently shorted (revealed by a premature closure of the local relay when the jumper is applied) and the proper functioning of the switch given a high temperature. The first test must produce closure of the steam supply line isolation valves, but the remainder verify closure of only the local relay. As above, it is recommended that the operator stationed at HPCI panel in the control room reset the autoisolation seal-in between each temper-

ature sensor functional test so that the entire autoisolation circuit can be verified operational.

Procedure 8.M.2-2.5.4. HPCI Steam Line Low Pressure -

This procedure tests the HPCI steam line low pressure switches and the 1-out-of-2 taken twice logic which produces an autoisolation signal from them. Switches are disconnected from the steam supply line (which results in the contacts closing due to low pressure) in patterns which test the circuit logic. The first test produces closure of the steam supply line isolation valves. Since low steam line pressure does not cause a seal-in of the autoisolation circuit, all relays deenergize when the pressure switches are reconnected to the steam line. Therefore, the current test verifies the functioning of all the isolation logic relays for all combinations of logic without requiring the operator to reset the autoisolation circuit and no change is necessary.

While this procedure does not affect the HPCI autoisolation fault tree, it does contribute test caused downtime of the initiation function. In addition, a false signal from the low pressure sensors can result in disabling the initiation function. Finally, if the initiation logic tests are to be accomplished while the reactor is shutdown for refueling, the signal produced by the low pressure sensors must be disabled, since it overrides a HPCI initiation signal. (This is also an good option for the operator to

have in the event there is a false low pressure signal that disables the HPCI injection function.)

6.6.1 QUANTITATIVE ANALYSIS

The quantitative analysis of the autoisolation function periodic tests is very strongly influenced by three important facts revealed by the fault tree analysis, which are discussed in the following sections.

1) The autoisolation function has a relatively large potential for common cause failures.

2) Aside from combinations of common cause failures, only two cut sets contribute significantly to the unavailability of the autoisolation function.

3) Autoisolation tests affect the unavailability of the injection function as well as the autoisolation function.

Common Cause Effects

Because of the high degree of redundancy in this function nine potential common cause failures are modeled in the fault tree. Three of them account for the necessity of locating the temperature sensors in three separate rooms to detect steam line breaks. There is a probability that a sensor at one location can not detect a break at one of the other locations in time to initiate the safety function. In this analysis we assume a probability of 0.01 that the temperature sensors can not detect a steam line break because of their location. One accounts for a break which is not large

enough to trip the dP sensors. The others account for potential calibration errors or calibration drift.

A design which provides the necessary redundancy and diversity of sensors to overcome a 1% chance of failure due to location reduces the importance of individual sensor failures. If one designs against a 1% chance that a break will occur where the sensor can not detect a steam leak because of its location, he is assuming a minimum unavailability for that sensor. That probability tends to dominate the probability that sensor has failed during standby.

Important Cut Sets

The fault tree analysis revealed no single component and only 12 two component cut sets in the autoisolation fault tree. Of these, seven involve loss of power, which is monitored and consequently the unavailability is assumed quite low. An eighth pertains to the suppression pool, which is normally isolated during standby. The ninth and tenth contain a combination of common cause failures of all the temperature sensors plus and common cause failures of the dP sensors, which are judged to be primarily demand or human error in this function. The final two are the major contributors to the unavailability of the autoisolation function which are sensitive to test interval variations:

29, 31 - Coincident failure of the two steam line isolation valves, and

19, 21 - Coincident failure of the two autoisolation relays.

The testing intervals of the components in these cut sets will dominate the quantitative analysis of the periodic test policy for the autoisolation function.

The three autoisolation functional tests are currently accomplished monthly over a two day period. Because the initial test signal produces closure of the two isolation valves, every sensor test checks the functioning of all four components in the two important cut sets. However, because of their quick succession, the second two tests are performed before standby failures have had an opportunity to occur. Therefore, although the valves and relays are cycled a total of three times during the month, their periodic test interval is still 30 days.

The lower curves on Figure 6.19 compares the current sequential testing policy with one in which the sensor tests are staggered. These curves give the unavailability of the autoisolation function verses the periodic test interval of the sensor tests. When the sensor tests are accomplished sequentially, the relays and valves are tested at the sensor testing interval. In the staggered tests the isolation relays and valves are tested at one third the interval shown. The staggered testing policy is superior to the sequential policy, because the relays and valves get tested at the staggering interval, while the less important cut

Unavailability of Autoisolation Function

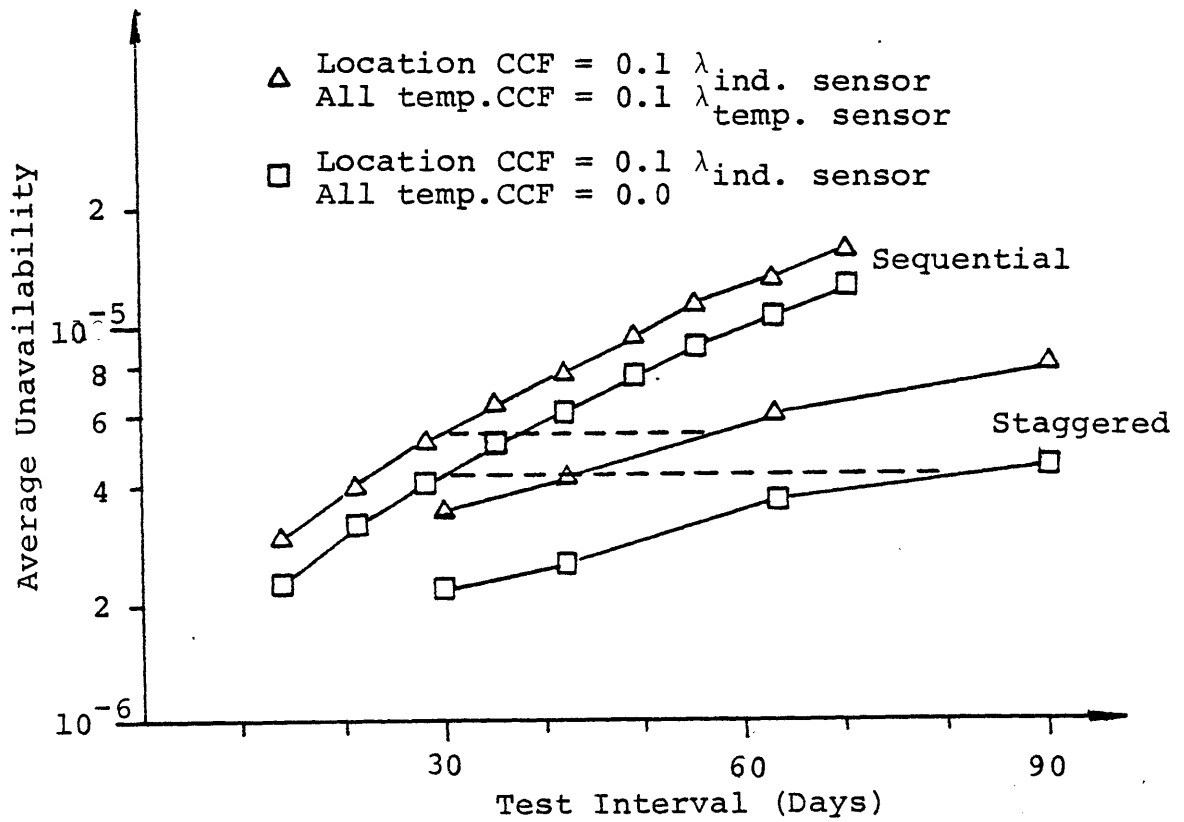


Figure 6.19. Autoisolation Function Unavailability as a Function of Autoisolation Sensor Test Intervals.

sets get tested less often. The decreased test interval for these dominant components reduces the function's unavailability.

The curves plotted with square data points are the result of assuming up to 10% dependent failures among sensors at any one location. The curves plotted with the triangular data points also assume that all the temperature sensors can fail with a $\lambda_{CCF} = 0.1\lambda_{ind}$. Because this assumption can defeat the designed redundancy of the temperature sensors, there is a larger percentage rise in the staggered testing unavailability than the sequential testing. However, the clear advantage of sequential testing is still evident.

It can be seen in the figure that, if we assume no common cause failures can fail all temperature sensors (the only plausible mechanism being calibration drift of all the sensors in one direction), the sensor tests can be accomplished once every nine weeks instead of at the current 30 day interval, with a reduction of 60% in the function's unavailability, because the 9 week sensor test interval translates into a 3 week test interval for the isolation relays and valves. Because an all temperature sensor λ_{CCF} increases the relative importance of them with respect to the valves and relays, the unavailability of staggered testing at 9 weeks is about the same as the current policy for that assumption.

Interaction With Injection Function

In this section it is assumed that staggered testing of the sensors is implemented. The autoisolation function tests interact with the injection function in two ways:

- 1) They close the Steam Line Isolation Valves and thus cause unavailability of the injection function.
- 2) They verify that the autoisolation sensors are not about to generate false autoisolation signals that would override the injection function.

When the two Steam Line Isolation Valves close during an autoisolation test, the HPCI System isolates from the reactor. While the valves are closed and the autoisolation signal is sealed-in the HPCI System can not respond to a true demand for the injection function unless an operator clears the test signal and resets the seal-in relay. The parameter q_0 accounts for the probability that these actions will not be accomplished. Because the conditions of a true demand would create a high stress situation with many coincident alarms, a Human Error Probability (HEP) of 0.25 is applied to each required action. [Sw80.]

The estimation of q_0 for the isolation valves is complicated by the fact that three different procedures are involved on a staggered basis. Since the HEP data and the times during individual tests when autoisolation signals are present are only rough estimates, it is reasonable to establish an average test time, τ , and unavailability to

override the tests, q_o , for the three tests. For brevity, only the estimation of the temperature sensor parameters is discussed.

According to plant personnel, the temperature test lasts approximately 3 hours. Much of that time is spent moving test equipment from sensor to sensor, so a test signal is applied to a temperature signal about 10% of the time. Given that our recommendation to clear to autoisolation signal after every test is implemented, the autoisolation signal is sealed in for another 10% of the time. During this time two operator actions are required: remove the signal and clear the seal-in. One is accomplished by the technician doing the test, and the second is accomplished by the control room operator stationed at the HPCI panel in accordance with the test procedures. During this time the control room operator must clear the seal-in signal before the HPCI can be initiated. During the remainder of the test the HPCI System could initiate its injection function should a true demand occur. Based on these estimates q_o for the temperature sensor test is calculated to be:

$$q_o = 0.1[1-(.75)(.75)] + 0.1(.25) = 0.07 \quad (6.4)$$

The first term reflects the fact that during 10% of the test two operators must perform actions successfully. The second term accounts for the time that only one operator must take action.

The other two sensor tests are shorter, but produce about the same q_o during the test as the temperature sensor test. When the contributions of all three procedures are averaged together, we obtain an average q_o of 0.08 over an average time of 1.5 hours per test.

The second cause of injection function unavailability due to autoisolation tests is failure of the isolation valves which requires that the HPCI System be declared inoperable for repair. The P_f will be equal to the probability that a valve fails to make a transition in either direction. This probability depends on the test interval and demand failures during both the opening and closing of the valves, yielding the relation:

$$P_f = \lambda T_2 + 2q_d \quad (6.5)$$

where $2q_d$ accounts for only transition failures which can cause occur under test conditions. The 2 accounts for the fact that if the valve fails to either close or open during the test, the HPCI System would be declared inoperable until repairs are completed. (In other words command failure mechanisms and accident condition failure mechanisms would not be included in this parameter.) Assuming that transition failure account for 10% of observed failures at the periodic test, the values of P_f shown in Table 6.5 for 10, 14, 21 and 30 day valve test intervals.

In order to determine the effect of the autoisolation tests on the injection function, the applicable cut sets relating to false signals from the autoisolation sensors and the closure of the Steam Line Isolation Valves are extracted from the injection function fault tree and subjected to testing in accordance with the staggered autoisolation test policy. Shorts in series sensors and excessive drift which could lead to false signals from those sensors are modeled by the sensor standby failure rate. Valve unavailability is created by the tests, as discussed above.

Figure 6.20 shows the unavailability contribution to the injection function due to autoisolation valve tests. It is interesting to note that, except for extremely high assumed common cause failure rates for generating false signals (ie. in excess of an individual component failure rate) the unavailability created in the injection function

| Sensor Test Interval (days) | Valve Test Interval (days) | P_f |
|-----------------------------|----------------------------|-------|
| 30 | 10 | .001 |
| 42 | 14 | .0013 |
| 63 | 21 | .0018 |
| 90 | 30 | .0023 |

Table 6.5. Probability of Test Caused Failure to Injection Function Failure Events 31 and 32 (Isolation Valves, NOFC) as a Result of Autoisolation Function Tests.

Unavailability of Injection Function

1 - $\lambda_{CCF} = 1 \times 10^{-6}/\text{hr}$

2 - $\lambda_{CCF} = 3 \times 10^{-7}/\text{hr}$

3 - $\lambda_{CCF} = 1 \times 10^{-7}/\text{hr}$

4 - $\lambda_{CCF} = 0$

5 - Unavailability due to isolation valve testing

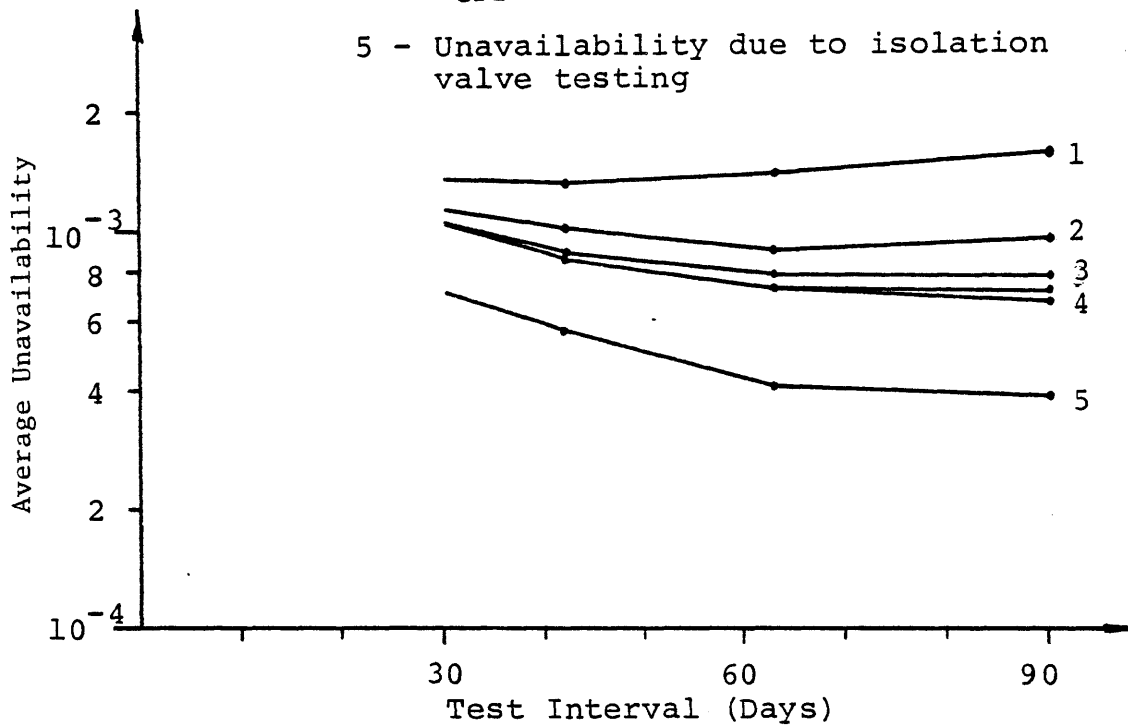


Figure 6.20. Injection Function Unavailability Due to Autoisolation Sensor Tests Which Cycle Autoisolation Valves MO 2301-4 and MO 2301-5.

by cycling the isolation valves is greater than the unavailability reduction obtained in the autoisolation function by verifying the lack of potential for false signals from the autoisolation circuit.

Recommendations

Figures 6.19 and 6.20 show that a staggered sensor testing policy can provide lower unavailability with longer individual sensor test intervals than the current policy, which schedules all sensor tests during a two day period of the month. Because the the two autoisolation valves, MO 2301-4 and 5, are cycled as a result of every sensor test, under a staggered testing policy they will be effectively tested more often. Since coincident valve and associated relay failures are the dominant two cut sets in the autoisolation function, it is possible to extend the individual sensor test intervals to either six or nine weeks and still reduce the autoisolation function's unavailability. Because the closure of the autoisolation valves during the sensor tests has the potential to defeat the injection function, the nine week test interval produces the least unavailability in the injection function and should be favored. At the nine week test interval the autoisolation

³ If the requirement to cycle MO 2301-4 and 5 is dropped from 8.5.4.4 (see Section 6.4.1) the requirements to cycle these valves will be reduced by a factor of 2.8 without a reduction in their unavailability.

valves will be cycled once every three weeks instead of the current three times per month.^{3 4}

The preceding calculations were made assuming a constant standby failure rate. To preclude the possibility that an increasing hazard rate might produce a sharper rise in the unavailability than calculated here, and consequently more out-of-range sensor failures, the test intervals for the three sensor tests could be first extended to 6 weeks with a staggering of two weeks between tests. If the amount of drift or number of failures observed in the sensors has not increased significantly, based on the observations of an operating cycle, then the sensor test interval may be increased to 9 weeks.

6.7 SUMMARY OF HPCI RECOMMENDATIONS

Table 6.6 summarizes the results of the quantitative assessment of the HPCI periodic testing policy.

As the injection function fault tree so clearly indicated, system unavailability is dominated by the failure of single component cutsets. Depending on assumptions regarding the failure rates of these components, the average unavailability of the system ranges from 0.01 to 0.03. One test, 8.5.4.1, the Turbine/pump and Valve Operability Test,

⁴ At the 6 and 9 week interval it may be desirable to calibrate the sensors every second test instead of every third.

| Procedure | Current Interval | Recommendations | Ref. Section |
|--|------------------|---|----------------|
| I. Injection Function Operability | | | |
| A. Online Tests | | | |
| 8.5.4.1 (Pump Flow Rate at 1000 psig) | Monthly | No Change | 6.4.3 |
| 8.5.4.4 (Valve Operability) | Monthly | 1) Delete Isolation Valves From Test 2) Consolidate Injection Valve Cycling Into 8.5.4.1 | 6.4.1 6.4.1 |
| 8.I.6 (Pump and Valve Operability) | Monthly | Consolidate into 8.5.4.1 | 6.4.1 |
| 8.A.15 (System Integrity Surveillance) | Quarterly | Consolidate into 8.5.4.1 | 6.4.1 |
| B. Refueling Cycle Tests | | | |
| 8.5.4.3 (Flow Rate at 150 psig) | Once/Cycle | No Change | 6.4.2 |
| 8.5.4.6 (Op. From Alt. Shutdown Stat.) | Once/Cycle | No Change | 6.4.2 |
| 8.E.23 (Flow Controller Calib.) | Once/Cycle | No Change | 6.4.2 |

Table 6.6. Summary of Periodic Test Recommendations

| Procedure | Current Interval | Recommendations | Ref. Section |
|---|------------------------------------|--------------------------------------|--------------|
| II. Initiation Function | | | |
| 8.M.2-2.1.1 (Reactor Water Level) | Func.: Monthly Calib. Quarterly | No Change | 6.5.1 |
| 8.M.2-2.1.4 (Dry Well Pressure) | Func.: Monthly Calib. Quarterly | No Change | 6.5.1 |
| 8.M.2-2.10.4.2 (Initiation Logic) | Semi-annually (April + Oct.) | 1) Consolidate into one procedure | 6.5.2 |
| 8.M.2-2.10.4.3 (Steam Supply Isol. Valve) | Semi-annually (April + Oct.) | 2) Test activation of all components | 6.5.2 |
| 8.M.2-2.10.4.4 (Injection Valve Logic) | Semi-annually (April + Oct.) | 3) Test once per re-fueling cycle | 6.5.3 |
| 8.M.2-2.10.12 (Reactor High Water Trip Logic) | Semi-annually (Feb. + Aug.) | No Change | |

Table 6.6. Summary of Periodic Test Recommendations (continued)

| Procedure | Current Interval | Recommendation | Ref. Section |
|--|-------------------|------------------------------------|--------------|
| III. Autoisolation Logic and Function | | | |
| 8.M.2-2.5.1 (Steam Line High Flow) | Monthly | Once every nine weeks* | 6.6.1 |
| 8.M.2-2.5.3 (Steam Space High Temperature) | Monthly | Once every nine weeks* | 6.6.1 |
| 8.M.2-2.5.4 (Steam Line Low Pres.) | Monthly | Once every nine weeks* | 6.6.1 |
| | | * Stagger at three week intervals. | |
| 8.M.2-2.10.5 (Auto-isolation Logic) | Semi-annually | Delete | 6.6 |
| 8.M.2-2.5.8 (Sup. Pool Isolation Valve) | 1/Refueling Cycle | No change | |

Table 6.6. Summary of Periodic Test Recommendations (continued)

| Procedure | Current Interval | Recommendations | Ref. Section |
|--------------------------------------|---|-----------------|--------------|
| IV. Miscellaneous Tests | | | |
| 8.5.4.5 (HPCI System Inoperable) | Specifies tests of other systems whenever HPCI down | Not addressed | |
| 8.C.13 (LO,LC Valve Surv.) | Once every two weeks | No change | |
| 8.M.2-2.5.6 (Cond. Stor. Tank Level) | Func: Monthly Calib: Quarterly | Not addressed | |
| 8.M.2-2.5.7 (Sup. Chamber Level) | Func: Monthly Calib: Quarterly | Not addressed | |

Table 6.6. Summary of Periodic Test Recommendations (continued)

verifies that the most important single component cut sets are functioning. It is recommended that the test interval of this test be maintained at 30 days. This interval results in an unavailability that is very close to the minimum, and the potential for test caused failures and wear-out argue against shortening it to match the calculated optimum.

The test intervals for many other tests should be lengthened as indicated in Table 6.6. Compared to the pump and valve operability tests, these tests address minor contributors to system unavailability. In two cases the lengthening of the test interval will actually improve the availability of the HPCI system.

1) Currently, it is necessary to deenergize the major components of the HPCI System to accomplish the initiation logic tests. The fault tree analysis has shown where three minor modifications to the logic circuits eliminate a single component cut set and make the function reliable enough to test on a once per cycle basis. The deletion of the necessity to disable the system to accomplish an online test removes a second single cut set, with the result that the overall unavailability of the initiation function is reduced by a factor of 80.

2) Staggering the autoisolation sensor tests permits verification of the operability of the isolation valves more often with fewer tests of the sensors, which have been designed with sufficient redundancy to make individual sen-

sensor failures relatively unimportant. Because the isolation relays and valves dominate the autoisolation function's unavailability, the sensor testing interval can be doubled while still reducing the unavailability in the autoisolation function by 40%. Because autoisolation function testing interferes with the injection function, a staggered and lengthened testing policy for the sensors also produces a 10 to 20% smaller contribution of autoisolation function failures to the injection function unavailability. (It should be pointed out, however, that the autoisolation function contributes only about 1% of the injection function's unavailability.)

In addition to the recommendations for changes in test interval, a number of recommendations have been made to consolidate similar procedures into a single comprehensive procedure. Also, some needlessly repetitious procedures should be eliminated.

The implementation of these recommendations will bring the periodic testing policy for the HPCI System into a better balance with the actual contributors to the the system's unavailability with 47 fewer tests per year.

More general conclusions regarding the practical application of time dependent unavailability analysis to operational standby safety systems are presented in Chapter 7.

CHAPTER 7

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

A large amount of effort is expended annually to test standby safety systems at nuclear power installations. Since the purpose of these tests is to verify that the systems can perform their safety function when required, it is prudent to ask how well they do their job and if the interval at which they are accomplished is consistent with their potential for improving availability.

7.1 SUMMARY

The purpose of this thesis has been to demonstrate that time dependent unavailability analysis is a practical tool for assessing the periodic testing program of standby safety systems. To accomplish this objective the following tasks have been accomplished:

- 1) The input parameters to the FRANTIC II computer code have been interpreted from an engineering point of view and related to failure causes which can be identified in failure and maintenance reports. This will assist systems engineers using the code.

- 2) Based on the engineering interpretation, FRANTIC II has been modified to model time dependent failure rates and test caused wear-out or burn-in with more flexibility. The

offset time added to the failure parameters now allows the user to model a component failure rate that can vary from any one value to any other value following any one of an infinite family of time dependent curves. The offset time also allows the user to investigate the effects of long term wear-out due to both testing and accumulated standby time without running a calculation over the entire life of the system.

3) A capability to calculate the optimum test interval of a constant failure rate component subject to imperfect testing has been added. The various contributors to imperfect testing are interpreted in terms of an effective downtime per test.

4) Subroutines obtained from UNRAC [Ka80] have been modified to produce and permanently store cut sets based on a fault tree of the safety function. The subroutines also evaluate these cut sets to produce instantaneous system unavailability in terms of the instantaneous component unavailabilites calculated by FRANTIC II. The resulting package is named FRANTIC II-MIT. It can be easily applied to any simple or complex system without the use of lengthy analytical formulas. The input necessary to generate the cut sets is sufficiently flexible to easily permit changes in fault tree logic resulting from system modifications. This makes the code an effective tool to use during both the design and operational phases of a system analysis.

5) FRANTIC II-MIT has been applied to simple systems to provide an understanding of the effects of various assumptions about failure mechanisms and component configurations. It was then used to assess the periodic testing program of the High Pressure Coolant Injection System of a Boiling Water Reactor. The resulting analysis led to recommendations for reducing system unavailability while also reducing the periodic test load from approximately 170 to 123 per year.

7.2 CONCLUSIONS

This thesis has provided the necessary background and examples to show that FRANTIC II-MIT can be used for practical applications in operational systems. It has demonstrated that time dependent unavailability analysis using FRANTIC II-MIT in the context of a comprehensive systems analysis can provide a quantitative basis for the establishment of periodic testing programs in standby safety systems. Based on our experience of interpreting, modifying, and applying the code, we make the following specific conclusions:

1) FRANTIC II-MIT can model essentially all significant component failure modes and mechanisms on a time dependent basis. It can be set up to answer a variety of system specific questions limited only by their capability to be represented by fault tree techniques and the data

available for code's wide variety of component failure parameters.

2) The application to the High Pressure Coolant Injection System has demonstrated FRANTIC II-MIT's potential for improving periodic test programs. The current policy requires over 20 different tests. FRANTIC II-MIT was able to model most situations and provided useful information upon which practical recommendations for test policy changes could be made.

3) Although precise input data is desirable, valuable insights about system behavior under testing can be gained through the use of FRANTIC II-MIT even with data having large uncertainties. By focusing on the potential for failure and its causes, a probabilistic approach such as used in this work can point out where those uncertainties make a difference. Often the relative change in a system's unavailability can provide the information needed to make test policy decisions. With sensitivity studies one can obtain this type of information with reasonable confidence despite uncertainties in the failure data.

4) FRANTIC II-MIT uses the computer efficiently. A typical problem with a number of calculations on a system of 10-20 components requires less than a minute of CPU time.

5) The effort required to properly analyze a periodic testing program in an operational system is extensive. A major part of it is associated with developing fault trees

and understanding operating procedures. FRANTIC II-MIT provides a valuable framework for gathering and using system information. It fits into the engineering process by allowing the designer to consider in detail the probability that a standby safety system will fail to accomplish its safety function. During this study many recommendations for improving test procedures came as a direct result of an attempt to obtain data to input to the code.

It should be mentioned that it is not necessary to use all the input parameters of FRANTIC II-MIT to analyze a fault tree. If only demand data are available the average probability of failure upon demand can be input as q_d and a quick estimate of system unavailability can be obtained from using just the one parameter.¹

6) The time dependent hazard rate models introduced in FRANTIC II-MIT had little impact on the results of the practical application of the code to the HPCI System, because of a lack of sufficient information to identify potential time dependencies in important components. Data supported the use of only the constant hazard rate model.

7) The ratio of demand related failures to standby failures is a very important factor to consider in develop-

¹ The original estimate of failure upon demand must account for the test interval. For example, WASH 1400 generally assumed a 30 day test interval in generating its failure upon demand data.

ing a periodic testing program. The application of FRANTIC II-MIT has shown that the system's unavailability becomes quite insensitive to test interval when demand failures become more probable than standby failures. Currently, failure rate data is not organized to allow the ready determination of the ratio of the two contributors.

7.3 RECOMMENDATIONS FOR FURTHER RESEARCH

1) FRANTIC II-MIT's failure mechanism modeling capabilities currently exceed the sophistication of the available failure data by a wide margin. It is recommended that a determination of the parameters useful in identifying failure mechanisms be made and a system for easily reporting this data in an operating environment be defined.

2) The generalized Weibull hazard rate introduced in FRANTIC II-MIT has the potential for application to a probability based maintenance policy where maintenance is accomplished to reduce a component's hazard rate. This requires work in two areas. First, research is necessary to determine how specific types of maintenance affect component hazard rates. Second, the component renewal type models in FRANTIC II-MIT need to be made more general to allow the hazard rate to be changed by a maintenance action. The choice of options for this second task will depend on the results of the first.

3) Frequently the unavailability of a safety function depends on more than just one standby safety system. Periodic test programs must be planned to insure that the systems are not all highly unavailable at the same time. The analysis of diverse independent systems with a single fault tree can be quite expensive. Consequently it would be desirable to store statistically independent system level instantantaneous unavailabilities for use in a higher level fault tree containing the systems and their common dependencies. The evaluation of such a fault tree requires modification of the TIMES subroutine of FRANTIC II-MIT to calculate system unavailability at specific user input time points as well as those generated by the periodic testing information. This will permit the evaluation of large fault trees in piecemeal fashion with considerable savings in computer time.

References

- Ap74. Apostolakis, George E., Mathematical Methods of Probabilistic Safety Analysis, PB-261 873, California Univ., Los Angeles, School of Engineering and Applied Science, Sept 1974. (Available from NTIS)
- Ap76. Apostolakis, R. G., "The Effect of a Certain Class of Potential Common Mode Failures on the Reliability of Redundant Systems", Nuclear Engineering and Design, 36 (1976), 123-133.
- Ap77. Apostolakis, G.E. and P.P. Bansal, "Effect of Human Error on the Availability of Periodically Inspected Redundant Systems," IEEE Transactions on Reliability R-26 (1977), 220-225.
- As68. Ascher, Harold E., "Evaluation of Repairable System Reliability Using the 'Bad-As-Old' Concept," IEEE Transactions on Reliability, R-17 (1968), 103-110.
- As78. Ascher, Harold E. and Harry Feingold, "Is There Repair After Failure?" Proceedings 1978 Annual Reliability and Maintainability Symposium, IEEE, 1978.
- As78. Ascher, Harold E. and Harry Feingold, "The Aircraft Air Conditioner Data Revisited," Proceedings 1979 Annual Reliability and Maintainability Symposium, IEEE, 1979, pp.153-159.
- As81. Ascher, Harold E., "Weibull Distribution vs 'Weibull Process'," 1981 Proceedings Annual Reliability and Maintainability Symposium, IEEE, 1981, 217-222.
- Ba75. Barlow, R. E. and F. Proschan, Statistical Theory of Reliability and Life Testing, Holt, Rinehart, and Winston, Inc, New York, 1975.
- Br73. Broberg, Henrik V. J., Inger A. M. Gustafson, and S. G. Fredrik Sandin, "Failure Rate Functions From Test Data," Proceedings 1973 Annual Reliability and Maintainability Symposium, 109-118.
- Br80a. Brune, R.L. and M. Weinstein, Development of a Checklist for Evaluating Maintenance, Test, and Calibration Procedures Used in Nuclear Power Plants, NUREG/CR-1368, SAND80-7053, HPT Inc., Thousand Oaks, CA, 1980.

References

- Br80b. Brune, R.L. and M. Weinstein, Procedures Evaluation Checklist for Maintenance, Test, and Calibration Procedures, NUREG/CR-1369, SAND80-7054, HPT, Inc., Thousand Oaks, CA 1980.
- Ca76. Caldarola, L., "A Method for the Calculation of the Cumulative Failure Probability Distribution of Complex Repairable Systems," Nuclear Engineering and Design, 36 (1976), 109-122.
- Ca77. Caldarola, L., "Unavailability and Failure Intensity of Components," Nuclear Engineering and Design 44 (1977), 147-162.
- Ch75. Chay, S.C. and M. Mazumdar, "Determination of Test Intervals in Certain Repairable Standby Protective Systems," IEEE Transactions on Reliability, R-24 (1975), 201-205.
- EP241. EPRI NP-241, "Assessment of Industry Valve Problems," MPR Associates, Inc., Electric Power Research Institute, Palo Alto, CA, November 1976.
- EP759. EPRI NP-759-WS, "Proceedings: Workshop on EPRI Availability Engineering," Electric Power Research Institute, Palo Alto, CA, March 1978.
- EP1064. EPRI NP-1064, "Analysis of Utility Industry Data Systems," Electric Power Research Institute, Palo Alto, CA, April 1979.
- EP1443. EPRI NP-1443 "Methodolgy (sic.) and Application of Probabilistic Evaluation to Thermal Reactor Safety," University of California, Electric Power Research Institute, Palo Alto, CA, July 1980.
- EP1558. EPRI NP-1558, "A Review of Equipment Aging Theory and Technology," Franklin Research Institute, Electric Power Research Institute, Palo Alto, CA, Sepy 1980.
- EP1570. EPRI NP-1570, VOLUME 1, "Verification of Vault Tree Analysis, Volume 1: Experiments and Results," Electric Power Research Institute, Palo Alto, CA, May 1981.
- Ep68. Epstein, Benjamin and Albert Schiff, Improving Availability and Readiness of Field Equipment Through Periodic Inspection, UCRL-50451, California Univ., Livermore. Lawrence Radiation Lab., July 16, 1968.

References

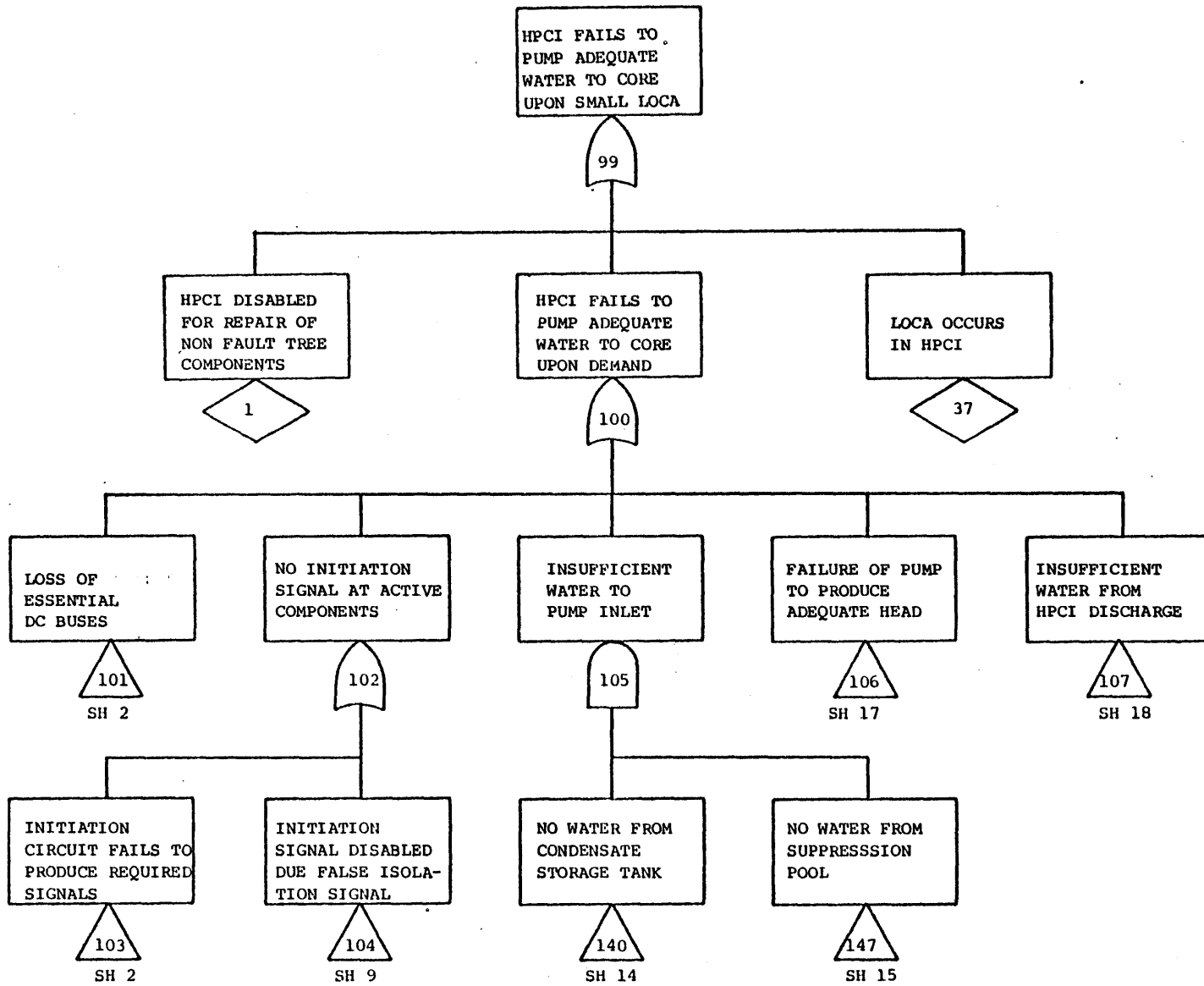
- Fu79. Fussell, J. B. and J. S. Arendt, "System Reliability Engineering Methodology: A Discussion of the State of the Art," Nuclear Safety 20 (1979), 541-550.
- GE80. NEDE-24809, "Probabilistic Analysis of the Reliability of BWR/4 Systems for Small LOCA Events," General Electric Company, 1980.
- Gr72. Green, A. E. and A. J. Bourne, Reliability Technology, New York: John Wiley and Sons, 1972.
- He81. Henley, Ernest J. and H. Kumamoto, Reliability Engineering and Risk Assessment, Englewood Cliffs, N.J.: Prentice-Hall, 1981.
- Hi71. Hirsch, Herbert M., "Setting Test Intervals and Allowable Bypass Times as a Function of Protective System Goals," IEEE Transactions in Nuclear Science, N-18 (1971), 488-494.
- IEEE500. IEEE Standard 500-1977, "IEEE Guide to the Collection and Presentation of Electrical Electronic and Sensing Component Reliability Data for Nuclear Power Generating Stations," IEEE, 1977.
- Ja68. Jacobs, I. M., "Reliability of Engineered Safety Features as a Function of Testing Frequency", Nuclear Safety, 9 (1968), 303-312.
- Ka80. Karimi, Roohollah, Qualitative and Quantitative Reliability Analysis of Safety Systems, Sc.D. Thesis, Massachusetts Institute of Technology, 1980.
- Ko74. Kontoleon, J. M., Nadia Kontoleon, and N. G. Chrysochoides, "Optimum Active-Inactive Times in Supervised Protective Systems for Nuclear Reactors," Nuclear Science and Engineering 55 (1974), 219-224.
- Ko78. Kontoleon, J. M., "Optimum Supervision Intervals and Order of Supervision in Nuclear Reactor Protective Systems," Nuclear Science and Engineering, 66 (1978), 9-13.
- Lo81. Lofgren, E., F. Varcolik, and W.E. Vesely, Optimum Test Intervals for Periodic Testing, NUREG/CR-2158, June 1981.
- Ma81. Mankamo, T., P. Aaltonen, P. Porvari and R. Virolsinen, "Allowable Repair Down-time in Stand-by Safety Systems," Paper 3.C.8, International ANS/ENS Top-

References

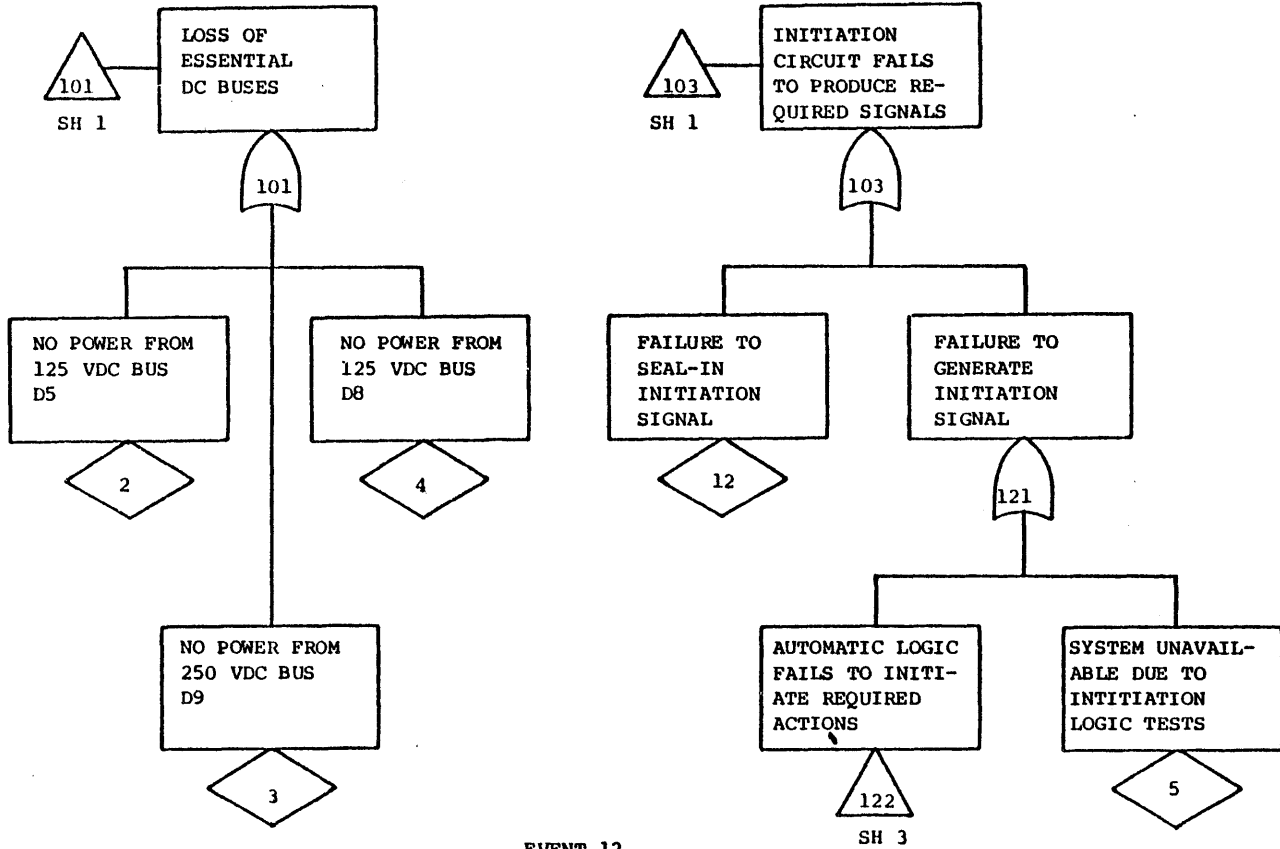
- ical Meeting on Probabilistic Risk Assessment, September 20-24, 1981.
- Ma82. Mankamo, Tuomas and Urho Pulkkinen, "Dependent Failures of Diesel Generators, Nuclear Safety, 23 (1982), 32-40.
- McC81. McCormick, Norman J., Reliability and Risk Analysis, Methods and Nuclear Power Applications. New York: Academic Press, 1981.
- McW80. McWilliams, T.P. and H. Martz, "Human Error Considerations in Determining the Optimum Test Interval for Periodically Inspected Standby Systems," IEEE Transactions on Reliability, R-29 (1980), 305.
- McW81. McWilliams, Thomas P. and Harry F. Martz, "Human Error Considerations and Annunciator Effects in Determining Optimal Test Intervals for Periodically Inspected Standby Systems," 1981 Proceedings Annual Reliability and Maintainability Symposium, 217-221.
- Mu72. Munford, A. G. and A. K. Shahani, "A Nearly Optimal Inspection Policy," Operational Research Quarterly, 23 (1972), 373-379.
- N2232. NUREG/CR-2232, Nuclear Plant Reliability Data System Annual Reports of Cumulative System and Component Reliability, Prepared by Southwest Research Institute, September 1981.
- Oelkers E. and W. W. Weaver, "The Impact of Aging Mechanisms on Reactor Safety System Performance," Nuclear Science and Engineering, 68 (1978), PP. 299-307.
- Ru68. Rubel, P., "Reliability of Reactor Components" Nuclear Safety 9 (1968), 481-486.
- Si79. Signoret, J.P., "Availability of Periodically Tested Systems," CEA-CONF 4830, 1979.
- Sw80. Swain, A.D. and H.E. Guttman, Handbook of Human Reliability Analysis With Emphasis on Nuclear Power Plant Applications, NUREG/CR-1278, April 1980.
- Th81. Thompson, Jr., W. A., "On the Foundations of Reliability," Technometrics, 23 (1981), 1-13.

References

- Va79a. Vaurio, Jussi K., "Unavailability of Components with Inspection and Repair," Nuclear Engineering and Design, 54 (1979) 309-324.
- Va79b. Vaurio, Jussi K. and D. Sciaudone, Unavailability Modeling and Analysis of Redundant Safety Systems, ANL-79-87, Argonne National Lab., Oct 1979.
- Va80. Vaurio, Jussi K., "Availability of Redundant Safety Systems with Common-Mode and Undetected Failures," Nuclear Engineering and Design, 58 (1980), 415-424.
- Va82. Vaurio, Jussi K., "Practical Availability Analysis of Standby Systems," Proceedings 1982 Annual Reliability and Maintainability Symposium, 125-130.
- Ve70. Vesely, W. E., "A Time-Dependent Methodology for Fault Tree Evaluation," Nuclear Engineering and Design, 13 (1970), PP. 337-360.
- Ve73. Vesely, W. E., "The Evaluation of Failure and Failure Related Data," Proceedings 1973 Annual Reliability and Maintainability Symposium, 500-506, Index serial no. 1113.
- Ve77. Vesely, W. E. and F. F. Goldberg, FRANTIC - A Computer Code for Time Dependent Unavailability Analysis, NUREG-0193, October 1977.
- Ve81. Vesely, W. E., F. F. Goldberg, J.T. Powers, et. al., FRANTIC II - A Computer Code for Time Dependent Unavailability Analysis, NUREG/CR-1924, April 1981.
- WASH1400. WASH-1400 (NUREG-75/014), "Reactor Safety Study, An Assessment of Accident Risks in U. S. Commercial Nuclear Power Plants," USNRC, 1975.
- Wo75. Wolf, L., "RE-BIT - A Computer Program for Fault Tree Analysis," Unpublished Work, Department of Nuclear Engineering, Massachusetts Institute of Technology, 1975.



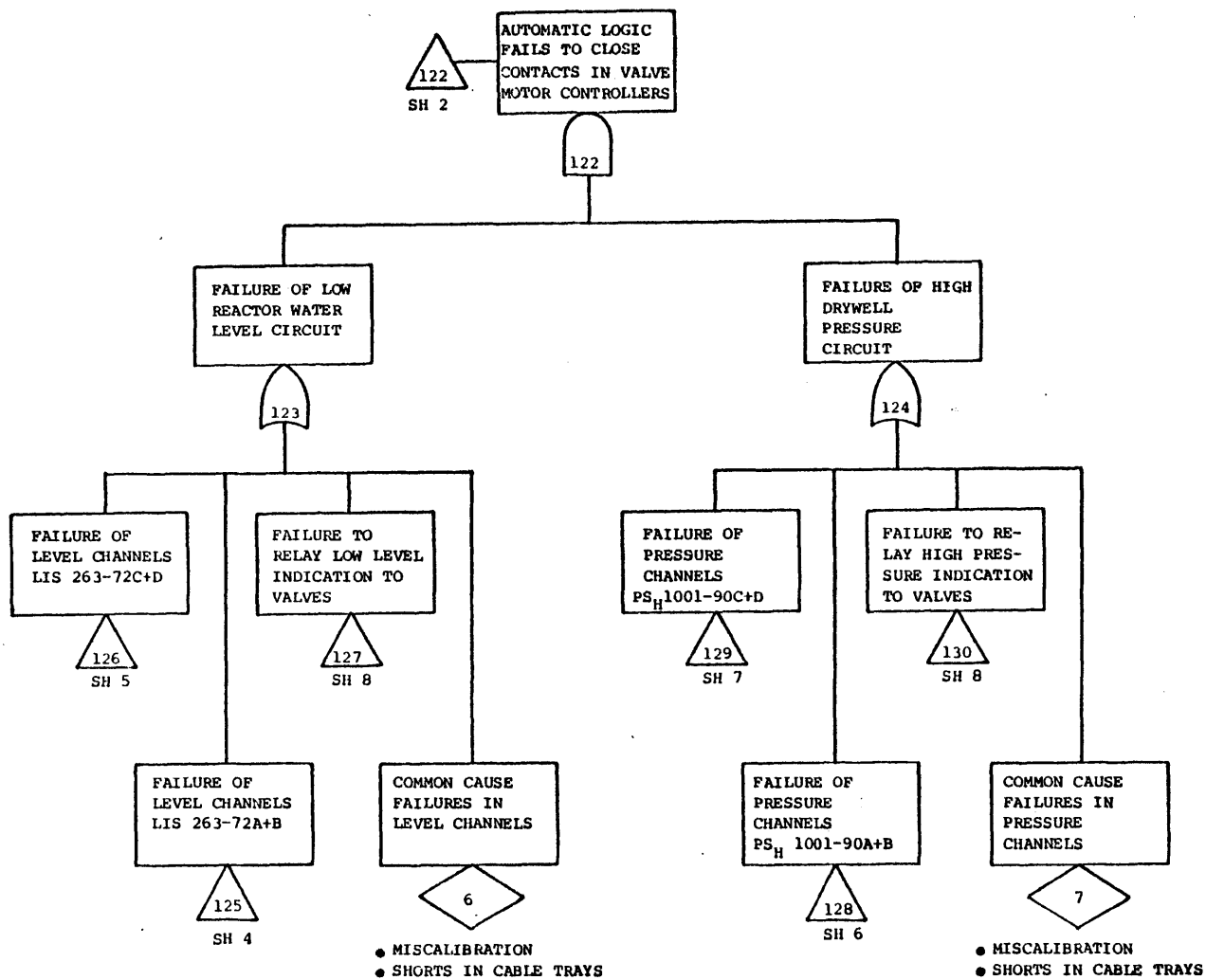
Appendix A. HPCI Injection Function Fault Tree
(continued)

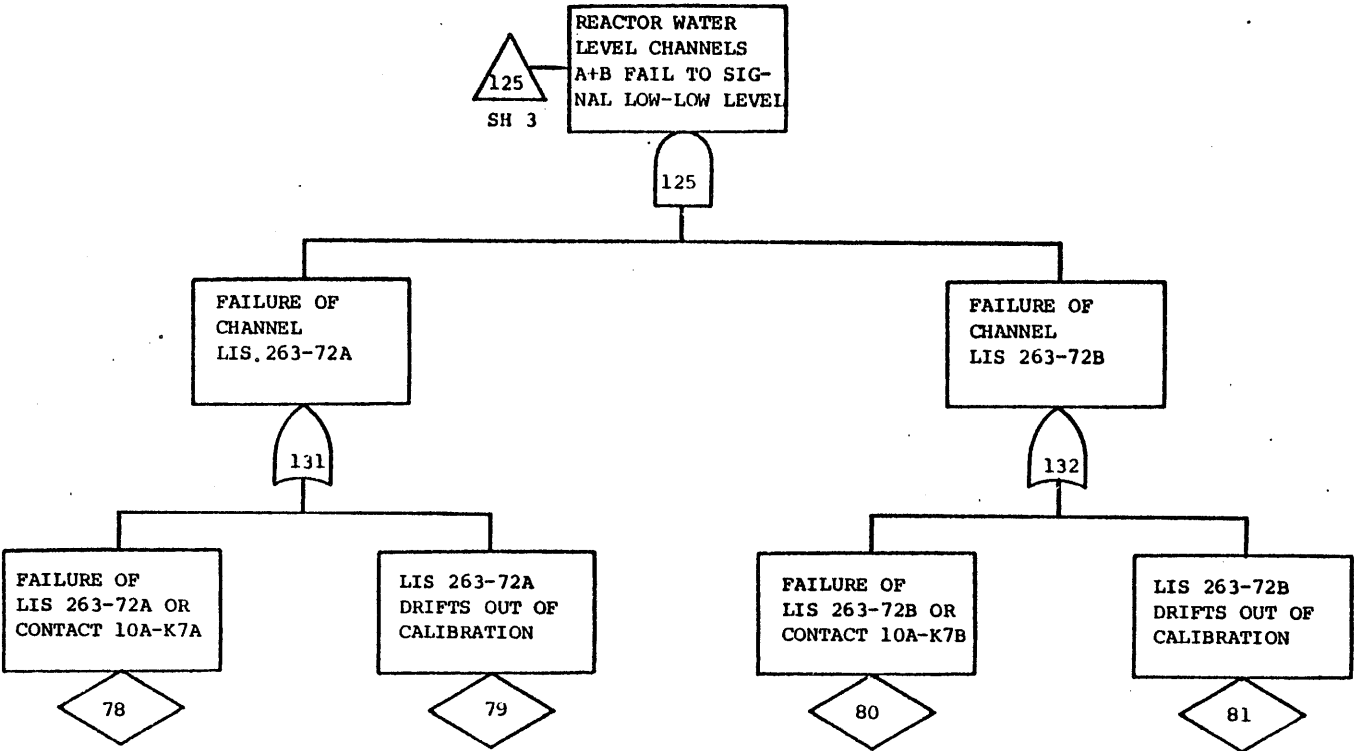


EVENT 12

- RELAY 23A-K23 FAILS TO ENERGIZE
 - RELAY 23A-K24 FAILS TO ENERGIZE
- (NOTE: MODIFICATION TO MAKE THESE RELAYS REDUNDANT IS RECOMMENDED. SEE SECTION 6.5 AND APPENDIX D)

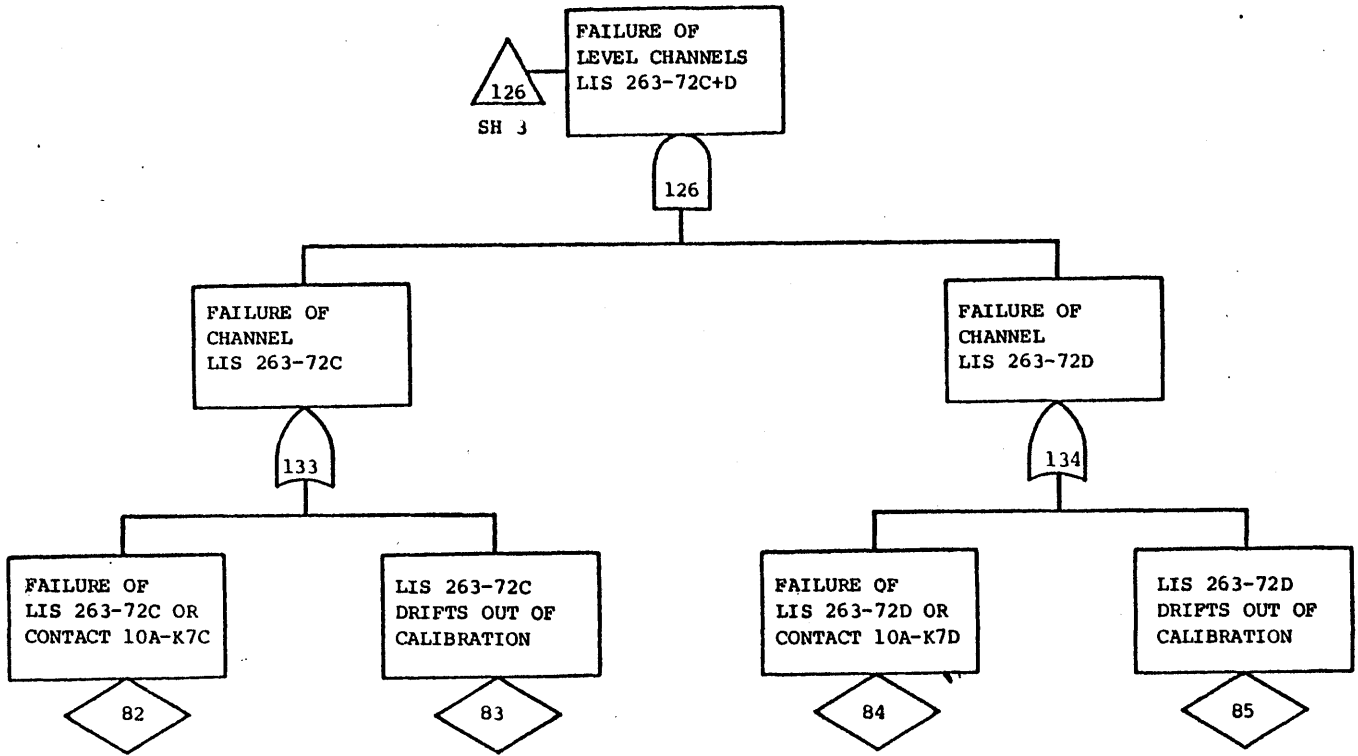
Appendix A. HPCI Injection Function Fault Tree
(continued)

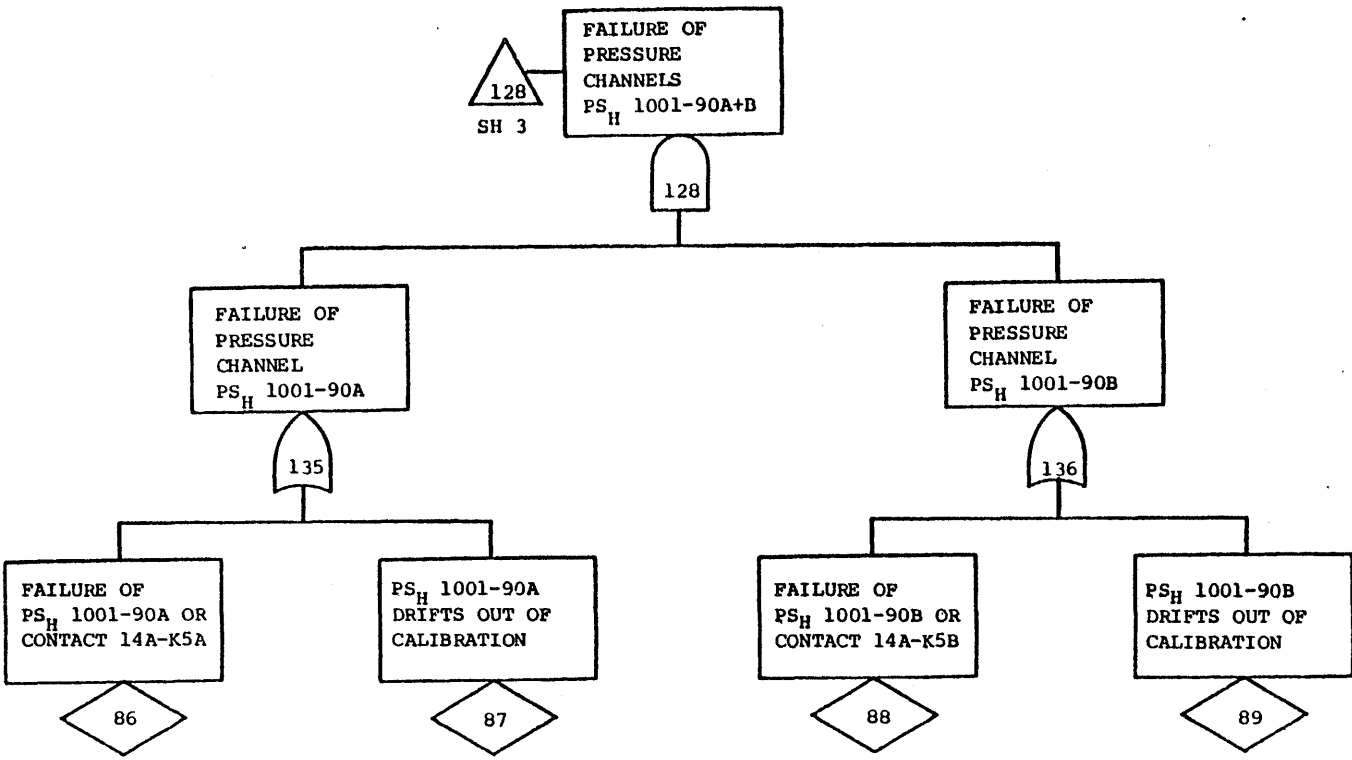




Appendix A. HPCI Injection Function Fault Tree
(continued)

Appendix A. HPCI Injection Function Fault Tree
(continued)

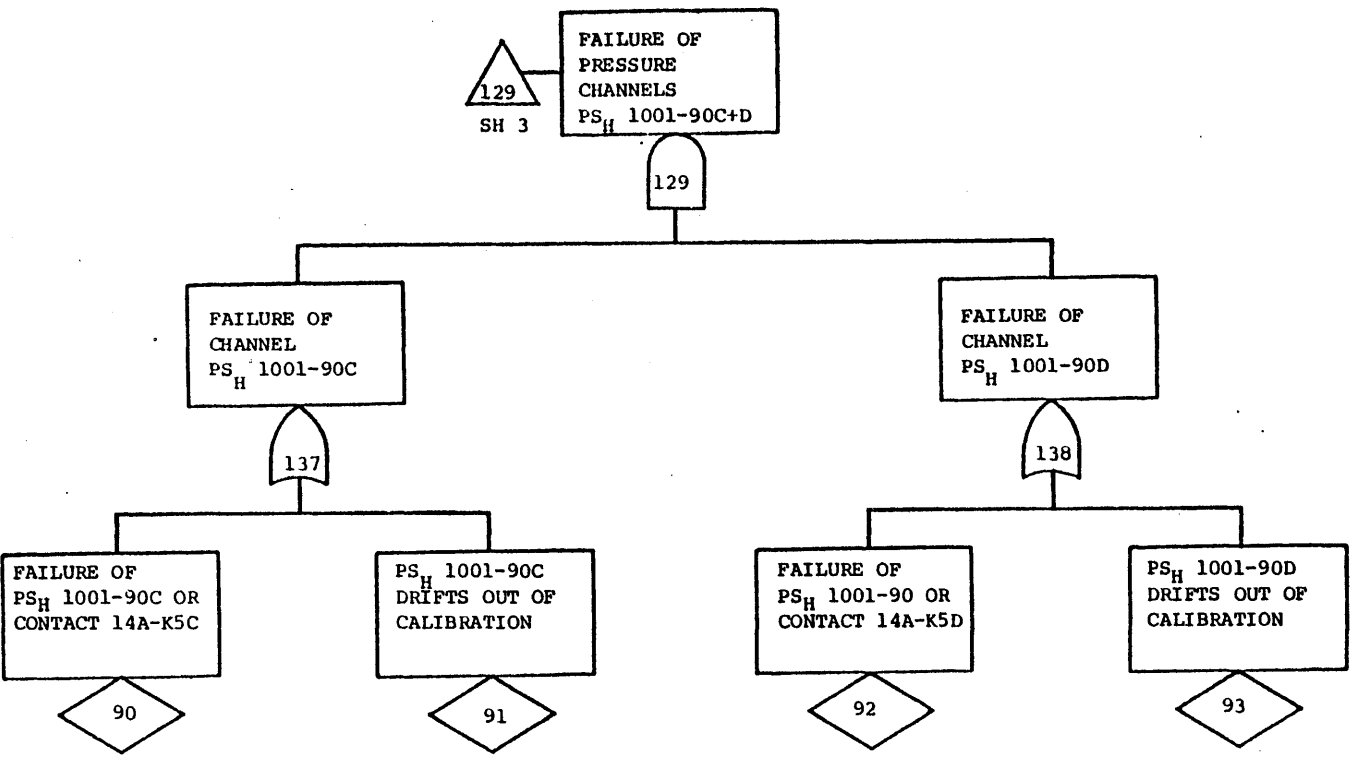




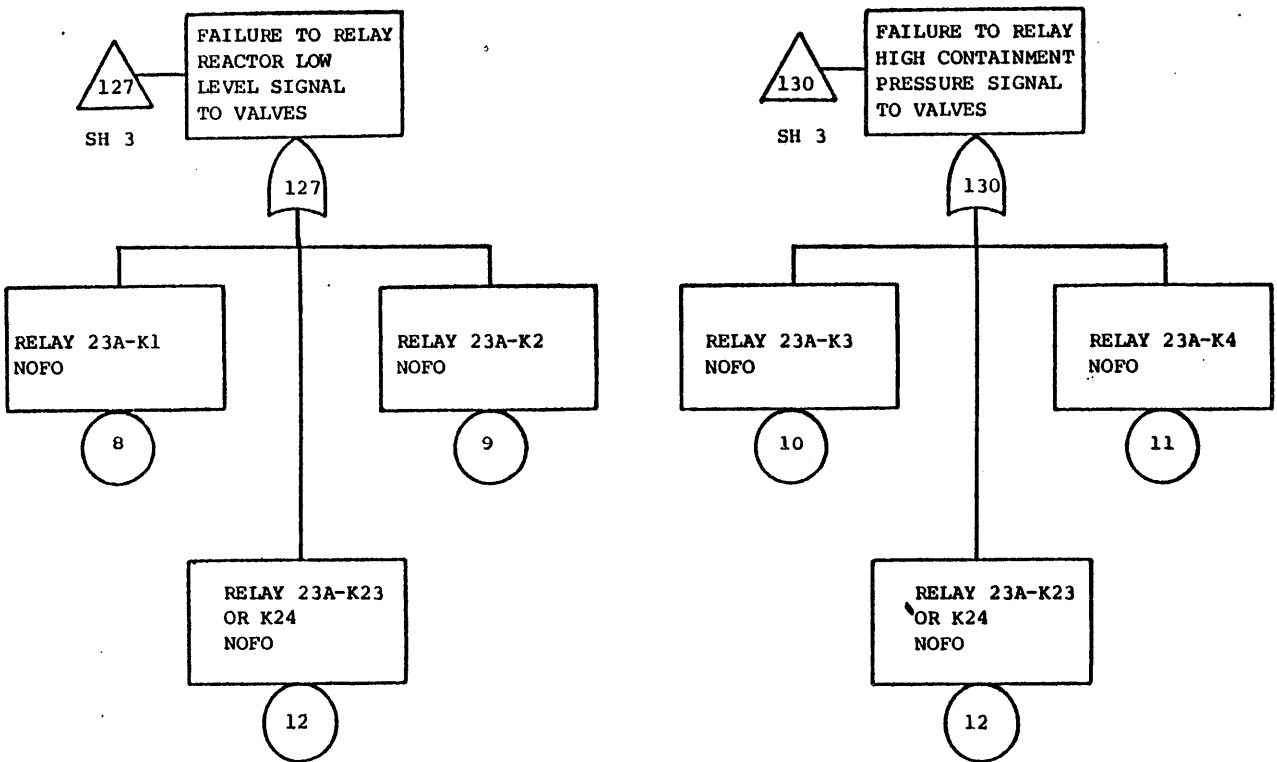
Appendix A. HPCI Injection Function Fault Tree
(continued)

4

Appendix A. HPCI Injection Function Fault Tree
(continued)

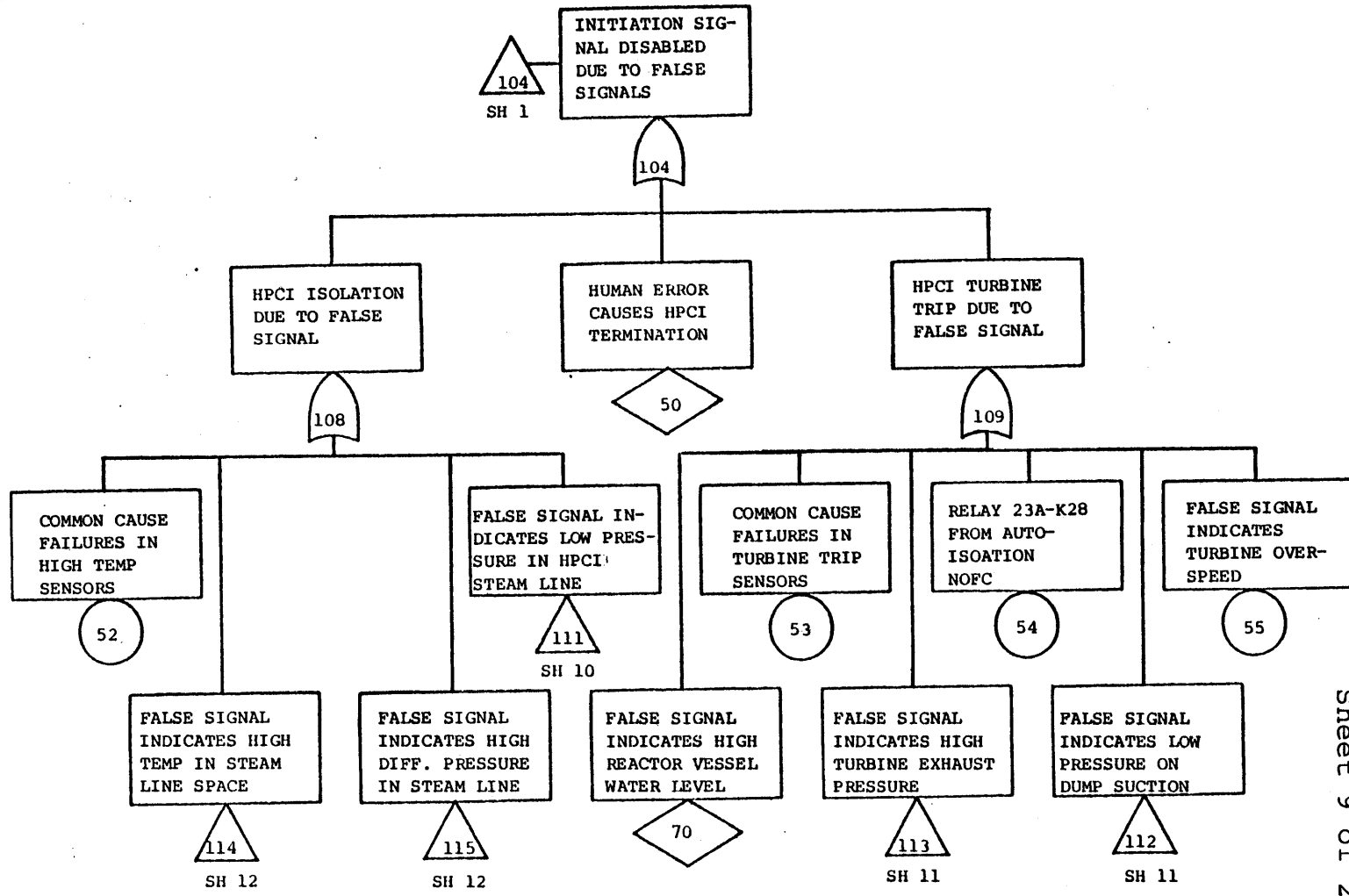


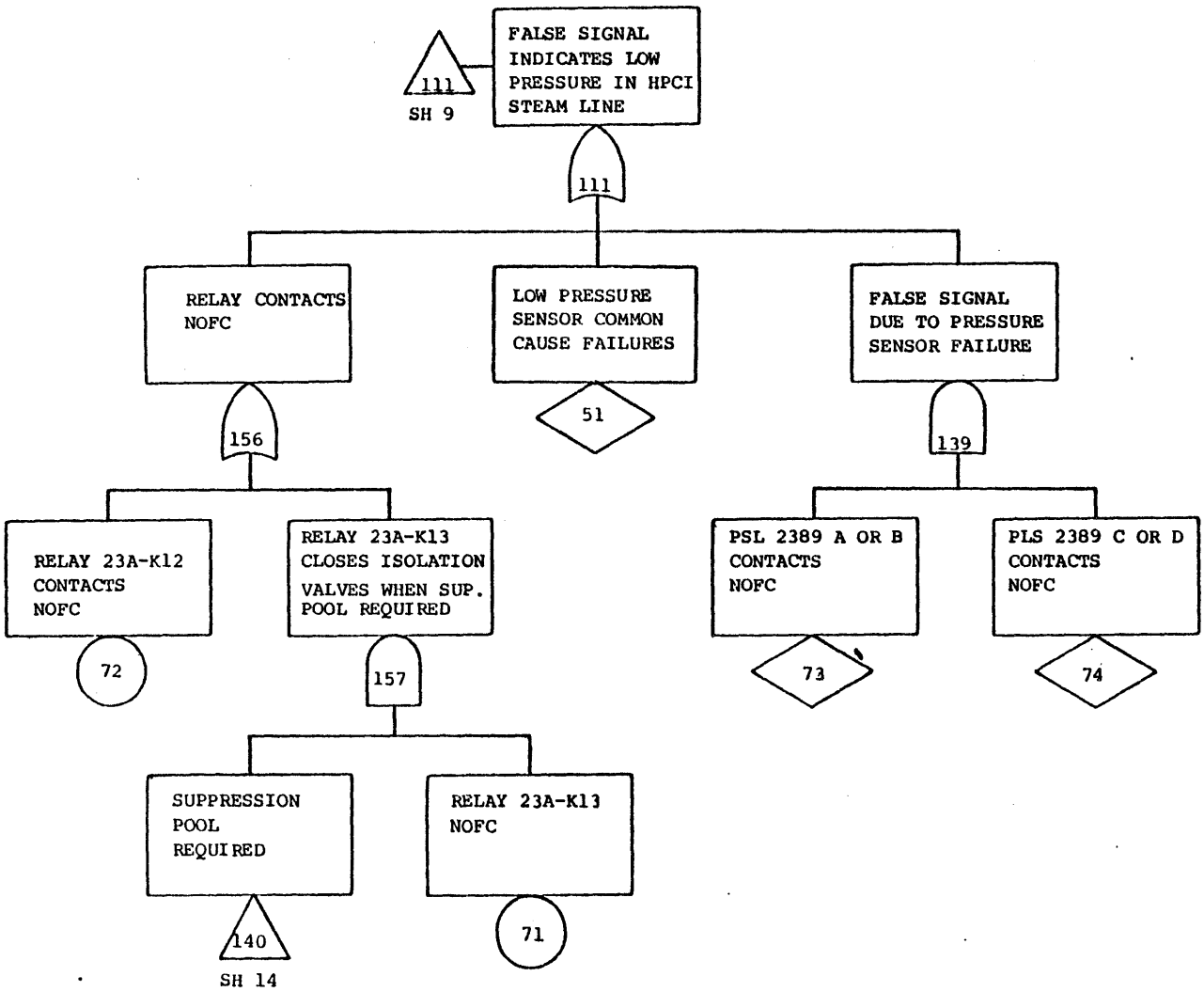
4



Appendix A. HPCI Injection Function Fault Tree
(continued)

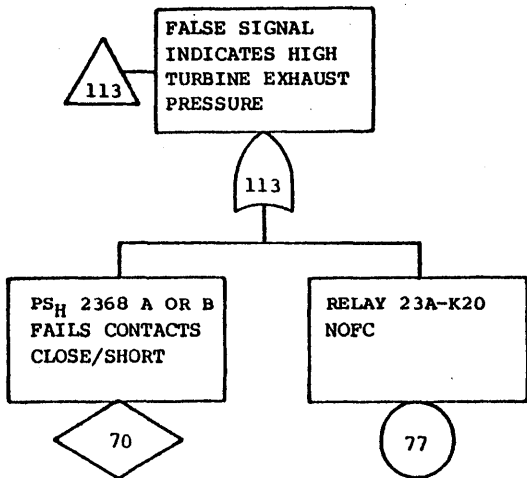
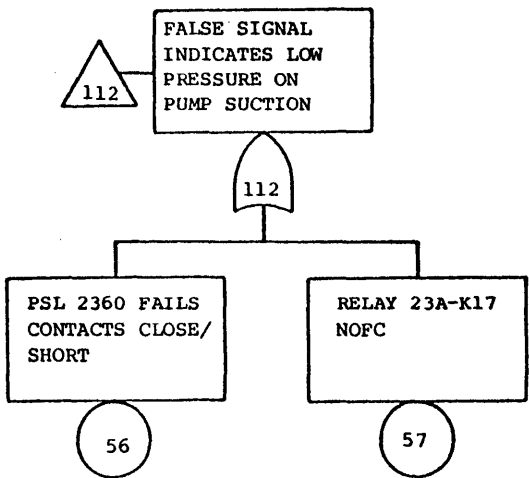
Appendix A. HPCI Injection Function Fault Tree
(continued)



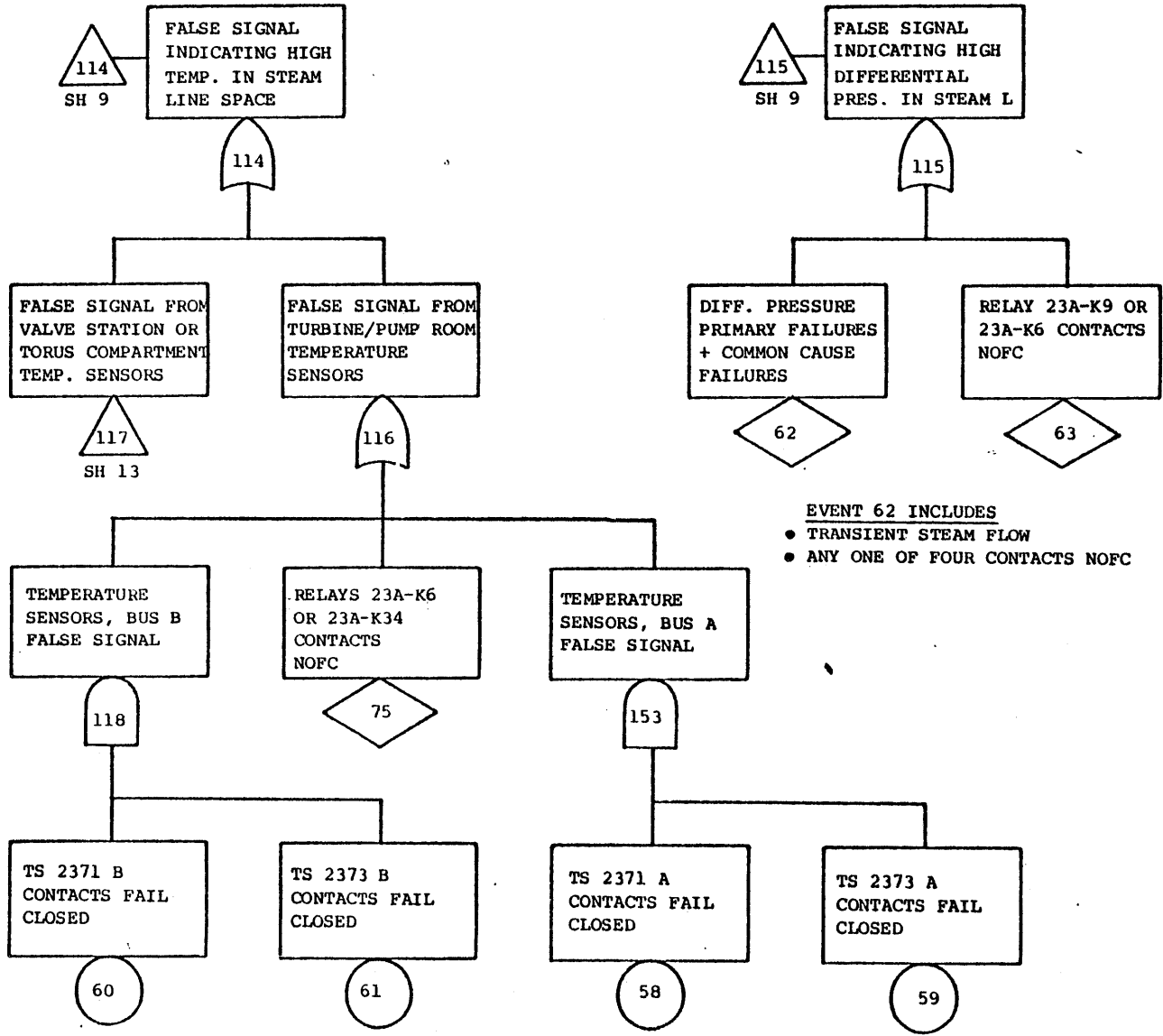


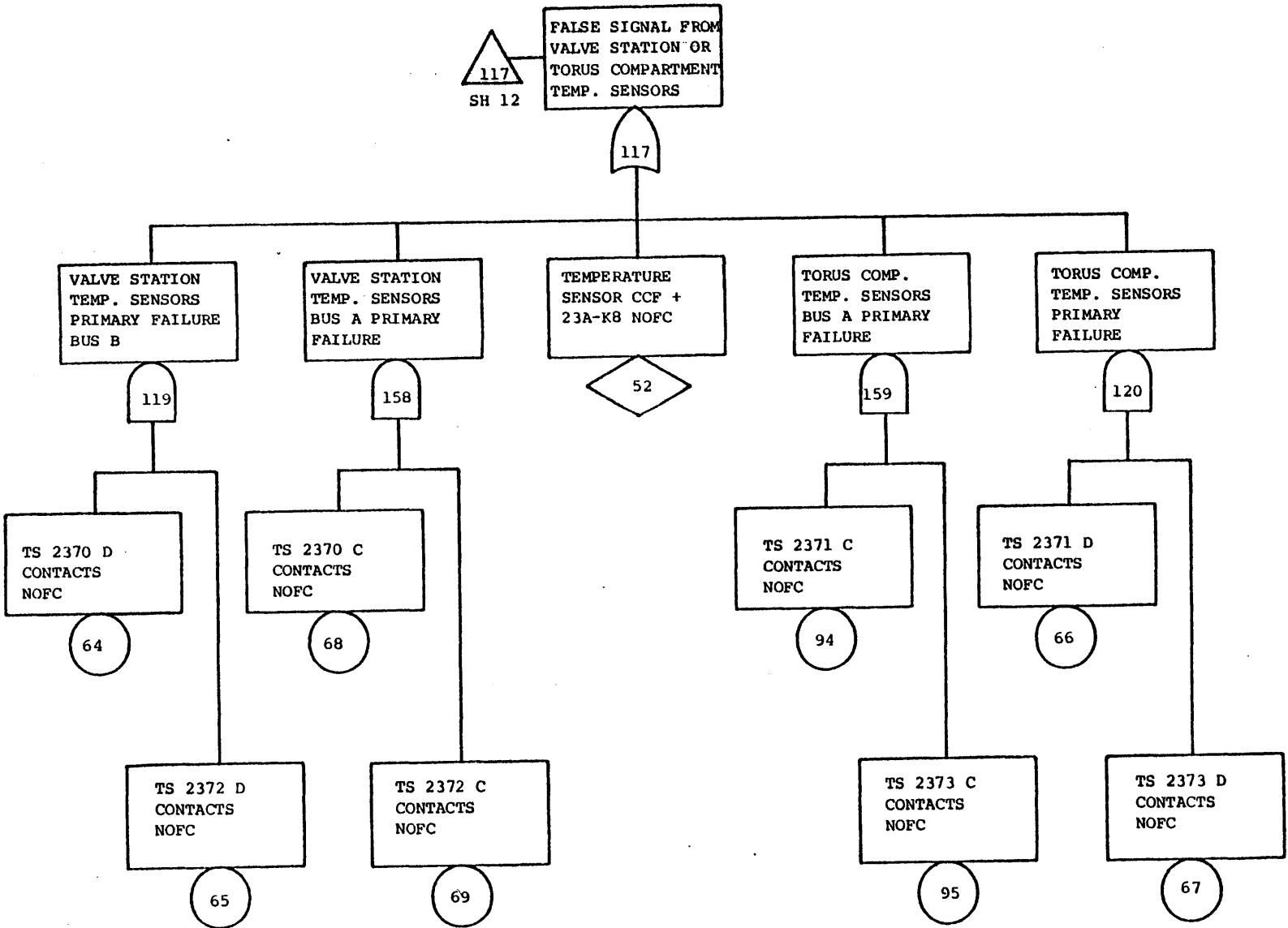
Appendix A. HPCI Injection Function Fault Tree
(continued)

Appendix A. HPCI Injection Function Fault Tree
(continued)

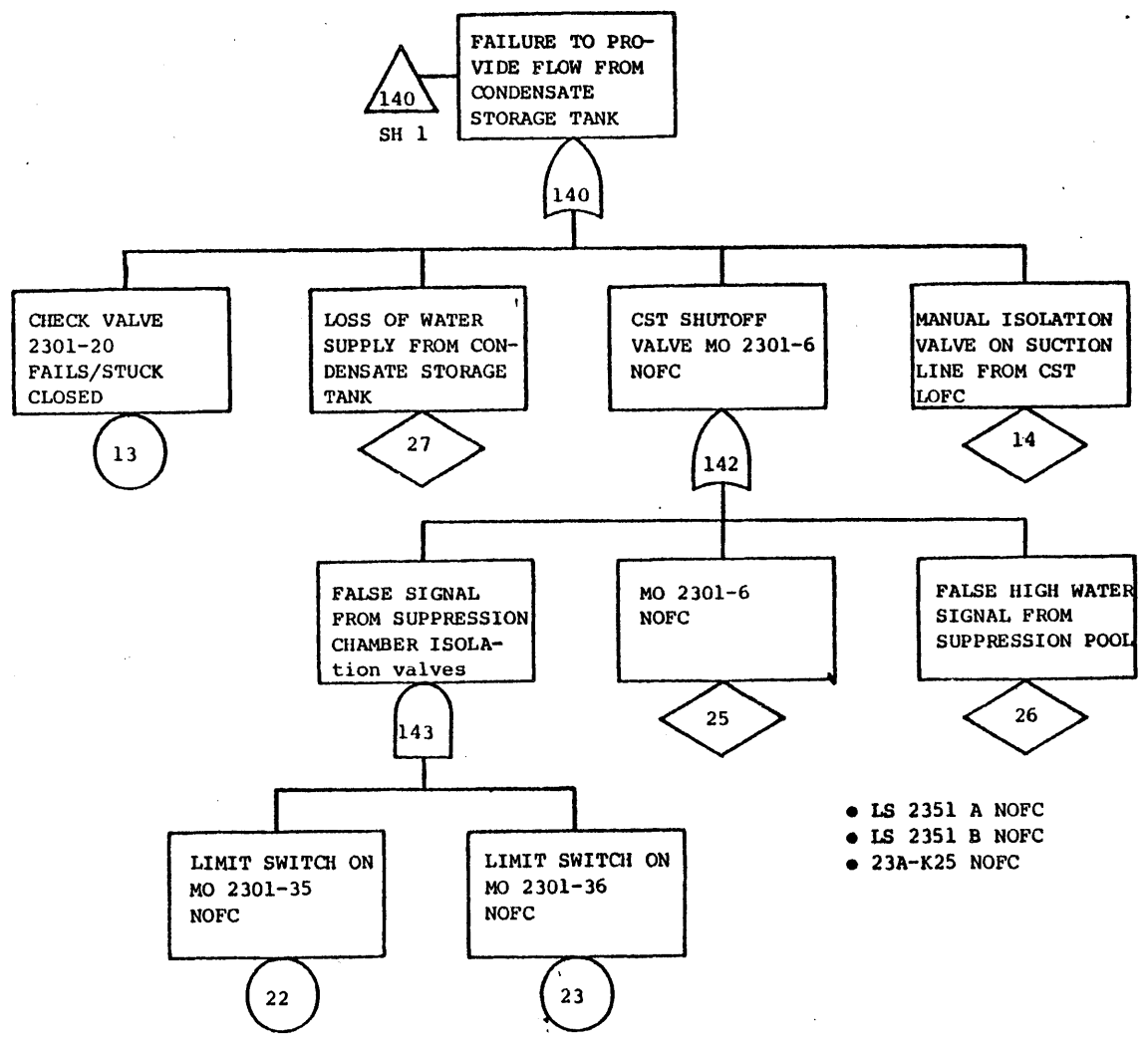


Appendix A. HPCI Injection Function Fault Tree
(continued)

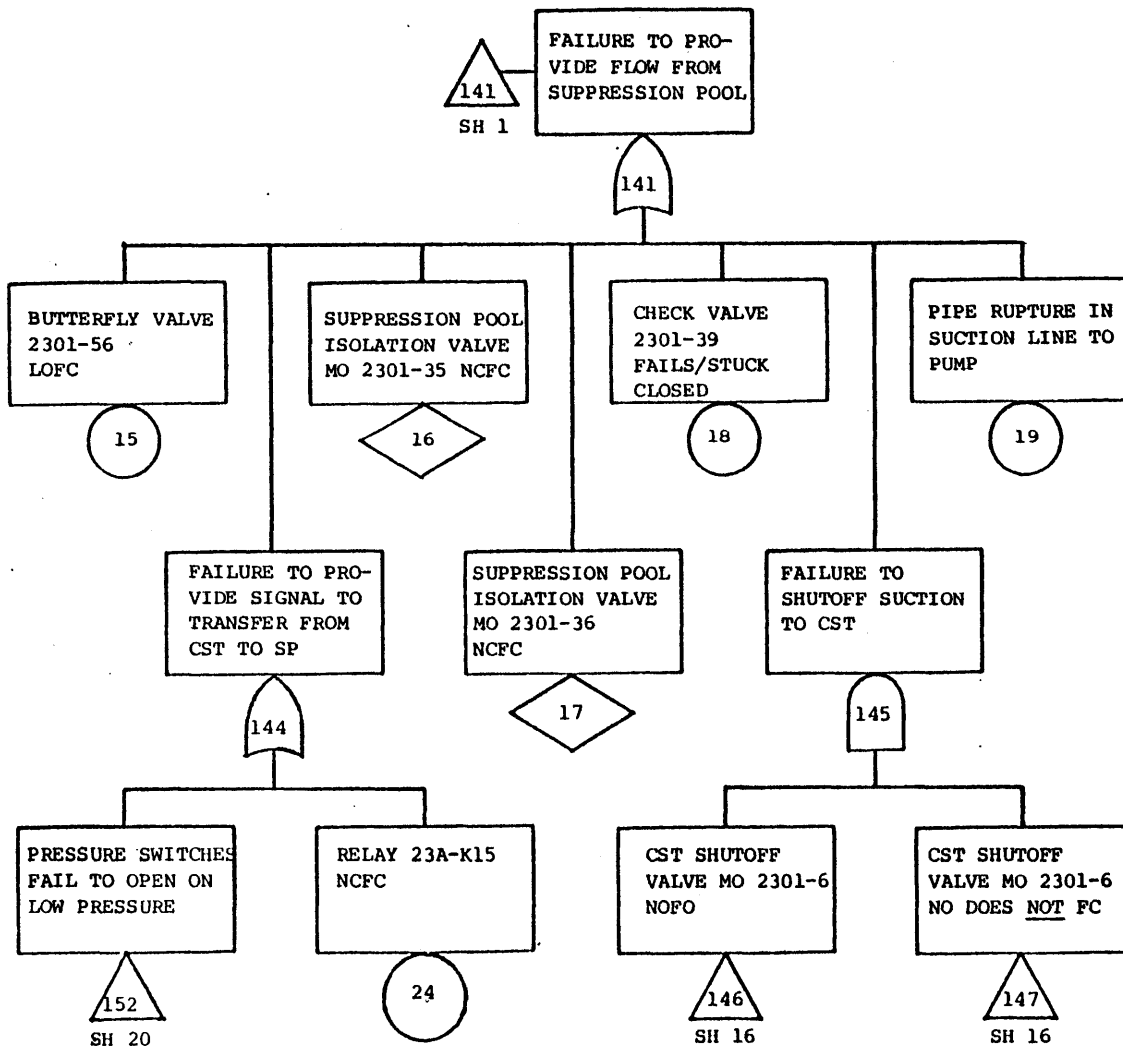




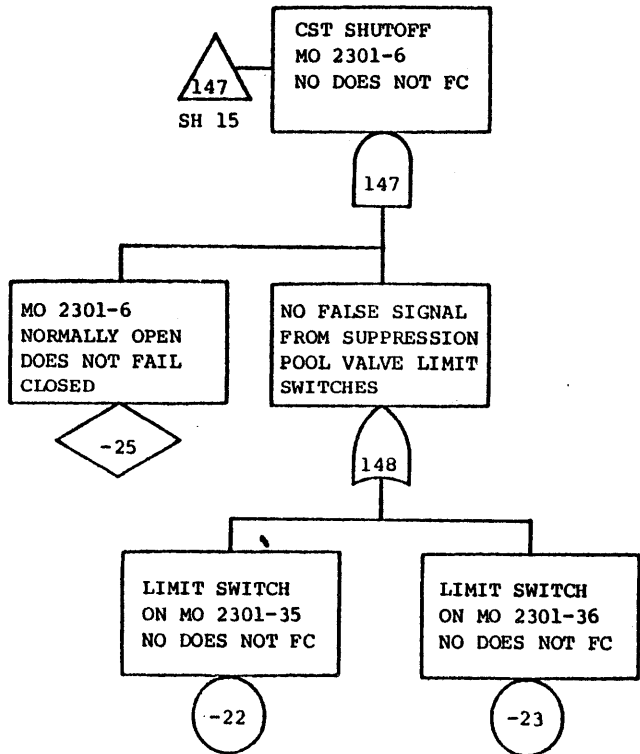
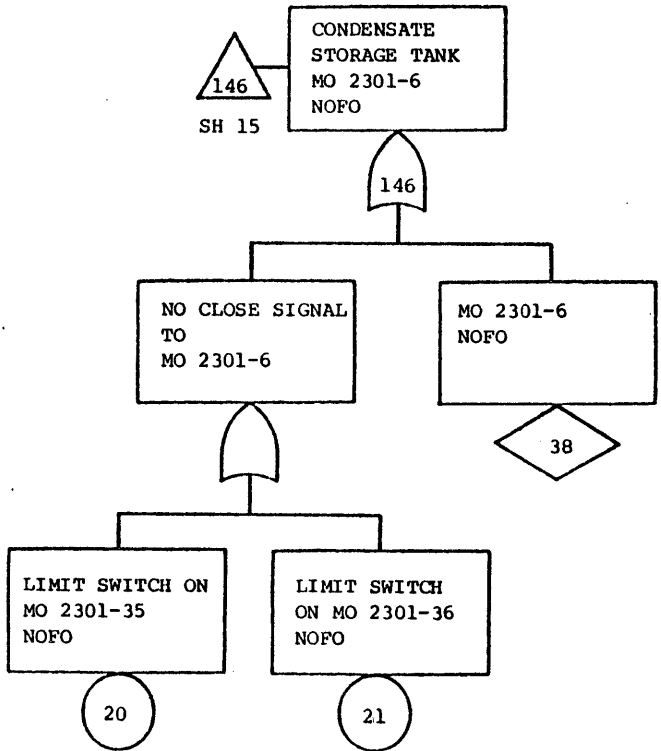
Appendix A. HPCI Injection Function Fault Tree (continued)



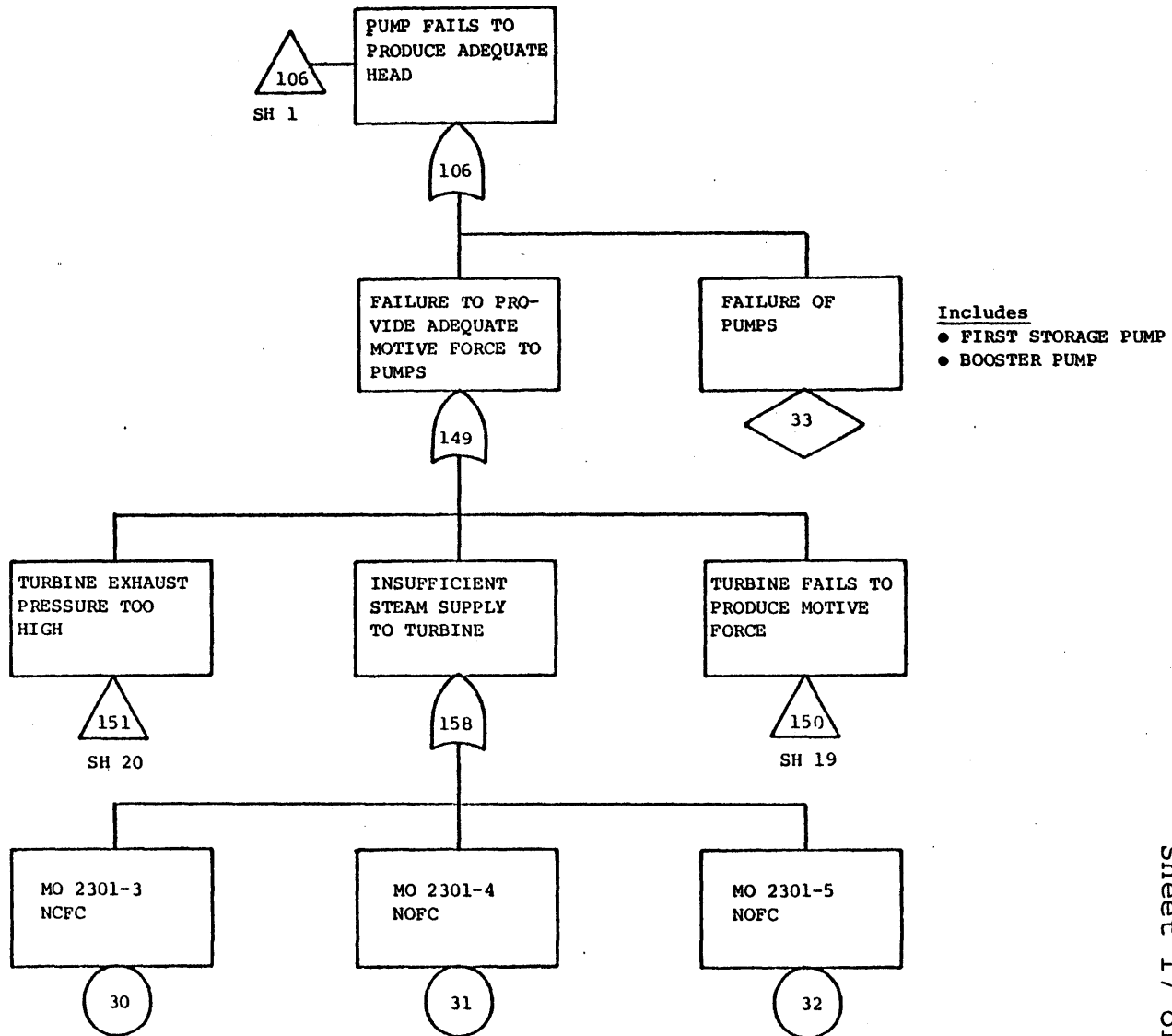
Appendix A. HPCI Injection Function Fault Tree
(continued)



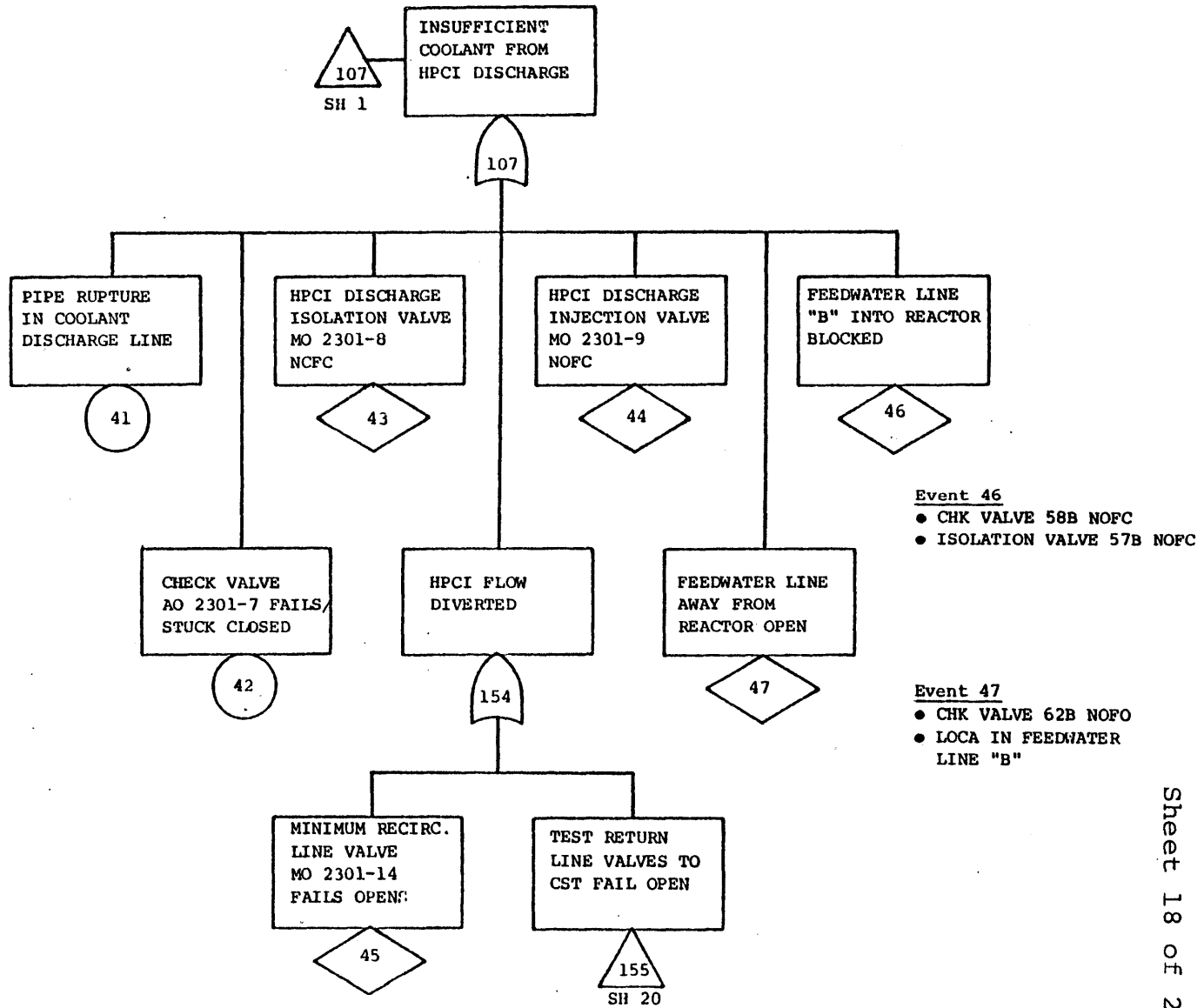
Appendix A. HPCI Injection Function Fault Tree
(continued)



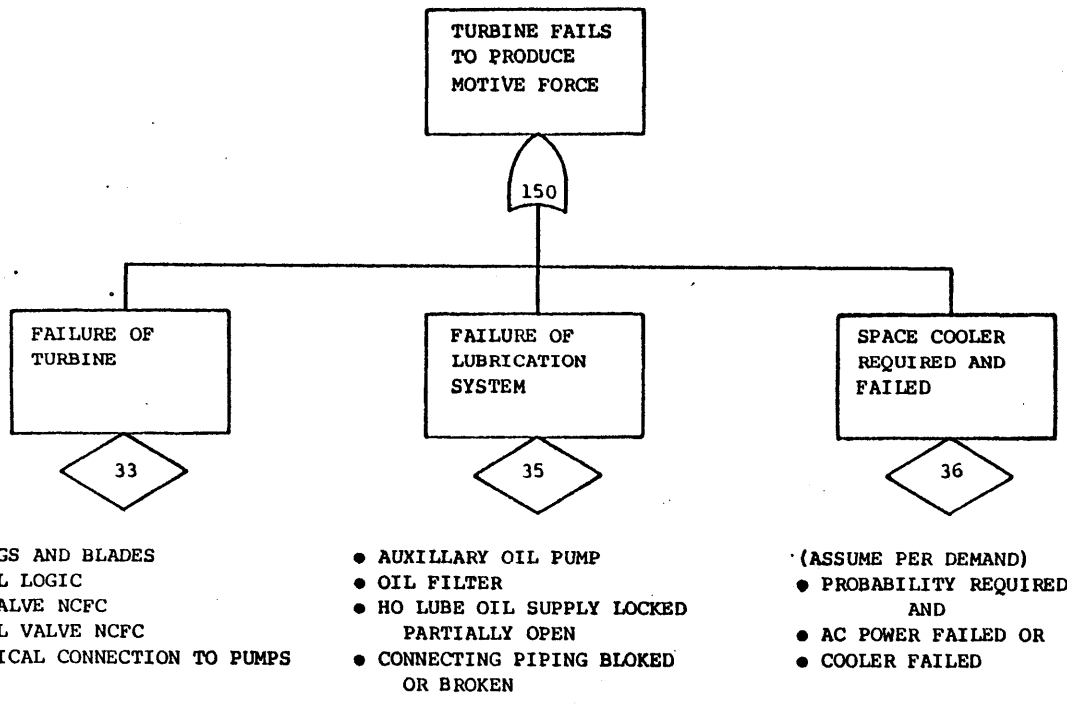
Appendix A. HPCI Injection Function Fault Tree
(continued)



Appendix A. HPCI Injection Function Fault Tree
(continued)



3



Appendix A. HPCI Injection Function Fault Tree
(continued)

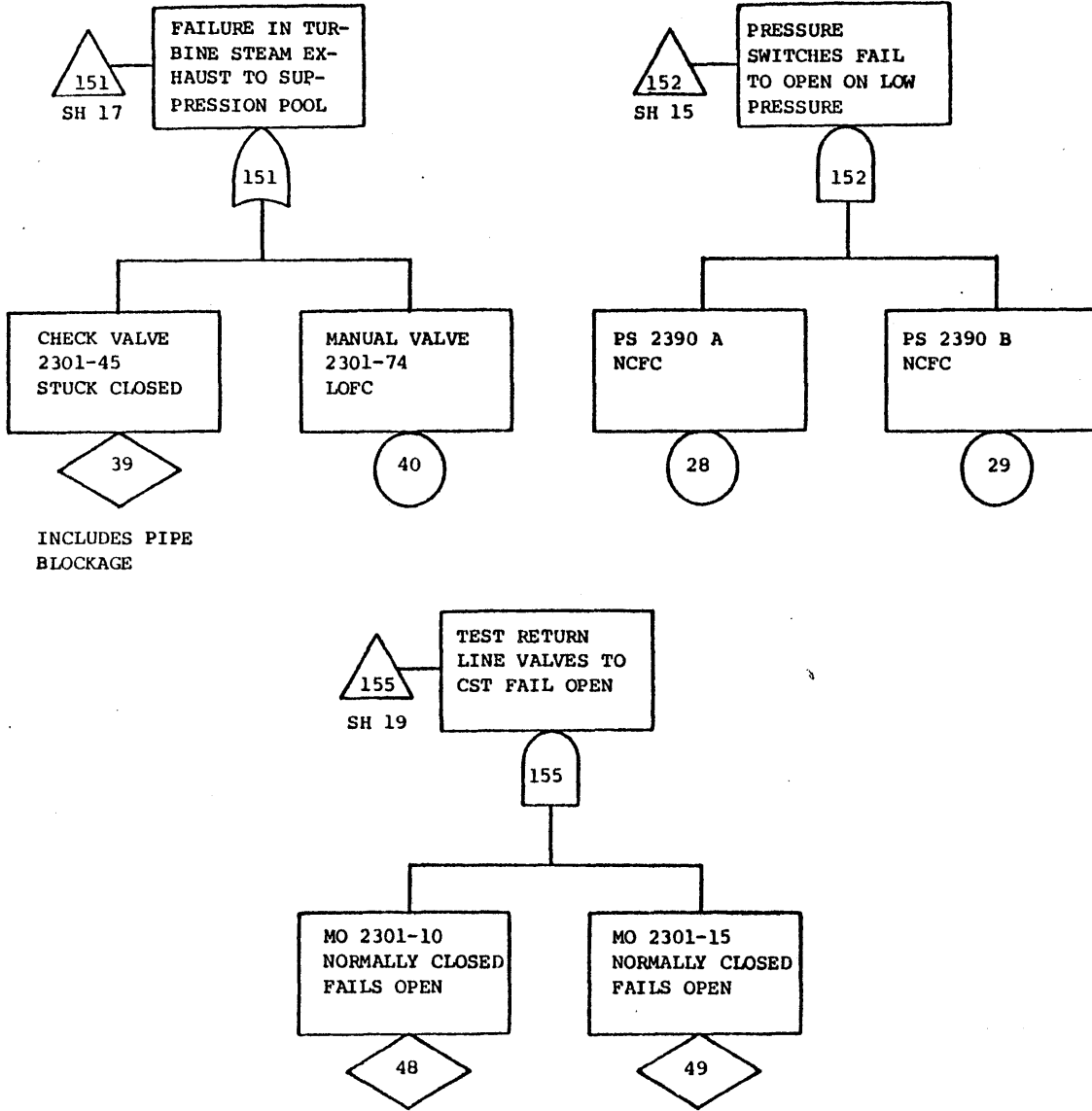


TABLE -1 FAULT TREE LOGIC

| GATE NO. | GATE TYPE | INPUT COMP. OR GATES | | | | | | | |
|----------|-----------|----------------------|-----|-----|-----|-----|----|----|---|
| 99 | 0 | 37 | 100 | 1 | 0 | 0 | 0 | 0 | 0 |
| 100 | 0 | 101 | 102 | 105 | 106 | 107 | 0 | 0 | 0 |
| 101 | 0 | 2 | 3 | 4 | 0 | 0 | 0 | 0 | 0 |
| 102 | 0 | 103 | 104 | 0 | 0 | 0 | 0 | 0 | 0 |
| 103 | 0 | 121 | 12 | 0 | 0 | 0 | 0 | 0 | 0 |
| 104 | 0 | 108 | 109 | 50 | 0 | 0 | 0 | 0 | 0 |
| 105 | 1 | 140 | 141 | 0 | 0 | 0 | 0 | 0 | 0 |
| 106 | 0 | 149 | 33 | 0 | 0 | 0 | 0 | 0 | 0 |
| 107 | 0 | 154 | 41 | 42 | 43 | 44 | 46 | 47 | |
| 108 | 0 | 114 | 115 | 52 | 111 | 0 | 0 | 0 | 0 |
| 109 | 0 | 112 | 113 | 53 | 54 | 55 | 70 | 0 | 0 |
| 110 | 0 | 114 | 115 | 0 | 0 | 0 | 0 | 0 | 0 |
| 111 | 0 | 139 | 156 | 51 | 0 | 0 | 0 | 0 | 0 |
| 112 | 0 | 56 | 57 | 0 | 0 | 0 | 0 | 0 | 0 |
| 113 | 0 | 77 | 70 | 0 | 0 | 0 | 0 | 0 | 0 |
| 114 | 0 | 116 | 117 | 0 | 0 | 0 | 0 | 0 | 0 |
| 115 | 0 | 62 | 63 | 0 | 0 | 0 | 0 | 0 | 0 |
| 116 | 0 | 118 | 75 | 153 | 0 | 0 | 0 | 0 | 0 |
| 117 | 0 | 119 | 120 | 158 | 159 | 76 | 0 | 0 | 0 |
| 118 | 1 | 60 | 61 | 0 | 0 | 0 | 0 | 0 | 0 |
| 119 | 1 | 64 | 65 | 0 | 0 | 0 | 0 | 0 | 0 |
| 120 | 1 | 66 | 67 | 0 | 0 | 0 | 0 | 0 | 0 |
| 121 | 0 | 122 | 5 | 0 | 0 | 0 | 0 | 0 | 0 |
| 122 | 1 | 123 | 124 | 0 | 0 | 0 | 0 | 0 | 0 |
| 123 | 0 | 125 | 126 | 127 | 6 | 0 | 0 | 0 | 0 |
| 124 | 0 | 128 | 129 | 130 | 7 | 0 | 0 | 0 | 0 |
| 125 | 1 | 131 | 132 | 0 | 0 | 0 | 0 | 0 | 0 |
| 126 | 1 | 133 | 134 | 0 | 0 | 0 | 0 | 0 | 0 |
| 127 | 0 | 8 | 9 | 12 | 0 | 0 | 0 | 0 | 0 |
| 128 | 1 | 135 | 136 | 0 | 0 | 0 | 0 | 0 | 0 |
| 129 | 1 | 137 | 138 | 0 | 0 | 0 | 0 | 0 | 0 |
| 130 | 0 | 10 | 11 | 12 | 0 | 0 | 0 | 0 | 0 |
| 131 | 0 | 78 | 79 | 0 | 0 | 0 | 0 | 0 | 0 |
| 132 | 0 | 80 | 81 | 0 | 0 | 0 | 0 | 0 | 0 |
| 133 | 0 | 82 | 83 | 0 | 0 | 0 | 0 | 0 | 0 |
| 134 | 0 | 84 | 85 | 0 | 0 | 0 | 0 | 0 | 0 |
| 135 | 0 | 86 | 87 | 0 | 0 | 0 | 0 | 0 | 0 |
| 136 | 0 | 88 | 89 | 0 | 0 | 0 | 0 | 0 | 0 |
| 137 | 0 | 90 | 91 | 0 | 0 | 0 | 0 | 0 | 0 |
| 138 | 0 | 92 | 93 | 0 | 0 | 0 | 0 | 0 | 0 |
| 139 | 1 | 73 | 74 | 0 | 0 | 0 | 0 | 0 | 0 |
| 140 | 0 | 142 | 13 | 14 | 27 | 0 | 0 | 0 | 0 |
| 141 | 0 | 144 | 145 | 15 | 16 | 17 | 18 | 19 | |
| 142 | 0 | 143 | 25 | 26 | 0 | 0 | 0 | 0 | 0 |
| 143 | 0 | 22 | 23 | 0 | 0 | 0 | 0 | 0 | 0 |
| 144 | 0 | 152 | 24 | 0 | 0 | 0 | 0 | 0 | 0 |
| 145 | 1 | 146 | 147 | 0 | 0 | 0 | 0 | 0 | 0 |
| 146 | 0 | 20 | 21 | 38 | 0 | 0 | 0 | 0 | 0 |
| 147 | 1 | 148 | -25 | 0 | 0 | 0 | 0 | 0 | 0 |

Appendix B. Injection Function Cut Sets

TABLE - 1 CONTINUED :

| GATE NO. | GATE TYPE | INPUT COMP. OR GATES | | | | | | | |
|----------|-----------|----------------------|-----|----|----|----|---|---|---|
| 148 | 0 | -22 | -23 | 0 | 0 | 0 | 0 | 0 | 0 |
| 149 | 0 | 150 | 151 | 30 | 31 | 32 | 0 | 0 | 0 |
| 150 | 0 | 34 | 35 | 36 | 0 | 0 | 0 | 0 | 0 |
| 151 | 0 | 39 | 40 | 0 | 0 | 0 | 0 | 0 | 0 |
| 152 | 1 | 28 | 29 | 0 | 0 | 0 | 0 | 0 | 0 |
| 153 | 1 | 58 | 59 | 0 | 0 | 0 | 0 | 0 | 0 |
| 154 | 0 | 155 | 45 | 0 | 0 | 0 | 0 | 0 | 0 |
| 155 | 1 | 48 | 49 | 0 | 0 | 0 | 0 | 0 | 0 |
| 156 | 0 | 157 | 72 | 0 | 0 | 0 | 0 | 0 | 0 |
| 157 | 1 | 140 | 71 | 0 | 0 | 0 | 0 | 0 | 0 |
| 158 | 1 | 68 | 69 | 0 | 0 | 0 | 0 | 0 | 0 |
| 159 | 1 | 94 | 95 | 0 | 0 | 0 | 0 | 0 | 0 |

Appendix B. Injection Function Cut Sets (continued)

TABLE - 2

| CUT SET NO. | NO. OF COMP. IN C. S. | COMPONENTS NOS. |
|-------------|-----------------------|-----------------|
| 1 | 1 | 37 |
| 2 | 1 | 2 |
| 3 | 1 | 1 |
| 4 | 1 | 34 |
| 5 | 1 | 3 |
| 6 | 1 | 4 |
| 7 | 1 | 12 |
| 8 | 1 | 5 |
| 9 | 1 | 33 |
| 10 | 1 | 39 |
| 11 | 1 | 30 |
| 12 | 1 | 31 |
| 13 | 1 | 32 |
| 14 | 1 | 35 |
| 15 | 1 | 36 |
| 16 | 1 | 41 |
| 17 | 1 | 42 |
| 18 | 1 | 43 |
| 19 | 1 | 44 |
| 20 | 1 | 46 |
| 21 | 1 | 47 |
| 22 | 1 | 45 |
| 23 | 1 | 56 |
| 24 | 1 | 50 |
| 25 | 1 | 62 |
| 26 | 1 | 52 |
| 27 | 1 | 75 |
| 28 | 1 | 40 |
| 29 | 1 | 77 |
| 30 | 1 | 53 |
| 31 | 1 | 54 |
| 32 | 1 | 55 |
| 33 | 1 | 70 |
| 34 | 1 | 57 |
| 35 | 1 | 63 |
| 36 | 1 | 51 |
| 37 | 1 | 76 |
| 38 | 1 | 72 |
| 39 | 2 | 73 74 |
| 40 | 2 | 64 65 |
| 41 | 2 | 15 22 |
| 42 | 2 | 58 59 |
| 43 | 2 | 8 10 |
| 44 | 2 | 7 8 |
| 45 | 2 | 6 10 |
| 46 | 2 | 6 7 |
| 47 | 2 | 13 15 |
| 48 | 2 | 13 16 |
| 49 | 2 | 13 17 |

Appendix B. Injection Function Cut Sets (continued)

TABLE - 2 CONTINUED :

| CUT SET NO. | NO. OF COMP. IN C. S. | COMPONENTS NOS. |
|-------------|-----------------------|-----------------|
| 50 | 2 | 13 18 |
| 51 | 2 | 13 19 |
| 52 | 2 | 13 24 |
| 53 | 2 | 14 15 |
| 54 | 2 | 14 16 |
| 55 | 2 | 14 17 |
| 56 | 2 | 14 18 |
| 57 | 2 | 14 19 |
| 58 | 2 | 14 24 |
| 59 | 2 | 15 27 |
| 60 | 2 | 16 27 |
| 61 | 2 | 17 27 |
| 62 | 2 | 18 27 |
| 63 | 2 | 19 27 |
| 64 | 2 | 24 27 |
| 65 | 2 | 15 25 |
| 66 | 2 | 15 26 |
| 67 | 2 | 15 23 |
| 68 | 2 | 16 25 |
| 69 | 2 | 16 26 |
| 70 | 2 | 16 23 |
| 71 | 2 | 17 25 |
| 72 | 2 | 17 26 |
| 73 | 2 | 17 23 |
| 74 | 2 | 18 25 |
| 75 | 2 | 18 26 |
| 76 | 2 | 18 23 |
| 77 | 2 | 19 25 |
| 78 | 2 | 19 26 |
| 79 | 2 | 19 23 |
| 80 | 2 | 24 25 |
| 81 | 2 | 24 26 |
| 82 | 2 | 23 24 |
| 83 | 2 | 16 22 |
| 84 | 2 | 17 22 |
| 85 | 2 | 18 22 |
| 86 | 2 | 19 22 |
| 87 | 2 | 22 24 |
| 88 | 2 | 22 71 |
| 89 | 2 | 66 67 |
| 90 | 2 | 68 69 |
| 91 | 2 | 94 95 |
| 92 | 2 | 60 61 |
| 93 | 2 | 9 10 |
| 94 | 2 | 9 11 |
| 95 | 2 | 8 11 |
| 96 | 2 | 7 9 |
| 97 | 2 | 6 11 |
| 98 | 2 | 48 49 |
| 99 | 2 | 13 71 |

Appendix B. Injection Function Cut Sets (continued)

TABLE - 2 CONTINUED :

| CUT SET NO. | NO. OF COMP. IN C. S. | COMPONENTS NOS. |
|-------------|-----------------------|-----------------|
| 100 | 2 | 14 71 |
| 101 | 2 | 27 71 |
| 102 | 2 | 25 71 |
| 103 | 2 | 26 71 |
| 104 | 2 | 23 71 |
| 105 | 3 | 6 86 88 |
| 106 | 3 | 10 82 84 |
| 107 | 3 | 7 82 84 |
| 108 | 3 | 8 90 92 |
| 109 | 3 | 10 78 80 |
| 110 | 3 | 9 86 88 |
| 111 | 3 | 9 86 89 |
| 112 | 3 | 8 87 88 |
| 113 | 3 | 8 86 89 |
| 114 | 3 | 6 90 92 |
| 115 | 3 | 7 78 80 |
| 116 | 3 | 6 87 88 |
| 117 | 3 | 6 86 89 |
| 118 | 3 | 11 78 80 |
| 119 | 3 | 11 78 81 |
| 120 | 3 | 10 79 80 |
| 121 | 3 | 10 78 81 |
| 122 | 3 | 7 79 80 |
| 123 | 3 | 7 78 81 |
| 124 | 3 | 25 28 29 |
| 125 | 3 | 26 28 29 |
| 126 | 3 | 23 28 29 |
| 127 | 3 | 11 82 84 |
| 128 | 3 | 11 82 85 |
| 129 | 3 | 10 83 84 |
| 130 | 3 | 10 82 85 |
| 131 | 3 | 7 83 84 |
| 132 | 3 | 7 82 85 |
| 133 | 3 | 9 90 92 |
| 134 | 3 | 9 90 93 |
| 135 | 3 | 8 91 92 |
| 136 | 3 | 8 90 93 |
| 137 | 3 | 8 91 93 |
| 138 | 3 | 9 91 92 |
| 139 | 3 | 9 87 88 |
| 140 | 3 | 8 87 89 |
| 141 | 3 | 6 91 92 |
| 142 | 3 | 6 90 93 |
| 143 | 3 | 13 28 29 |
| 144 | 3 | 6 91 93 |
| 145 | 3 | 6 87 89 |
| 146 | 3 | 11 79 80 |
| 147 | 3 | 10 79 81 |
| 148 | 3 | 7 79 81 |
| 149 | 3 | 14 28 29 |

Appendix B. Injection Function Cut Sets (continued)

TABLE - 2 CONTINUED :

| CUT SET NO. | NO. OF COMP. IN C. S. | COMPONENTS NOS. |
|-------------|-----------------------|-----------------|
| 150 | 3 | 27 28 29 |
| 151 | 3 | 22 28 29 |
| 152 | 3 | 8 86 88 |
| 153 | 3 | 11 83 84 |
| 154 | 3 | 10 83 85 |
| 155 | 3 | 7 83 85 |
| 156 | 3 | 9 91 93 |
| 157 | 3 | 9 87 89 |
| 158 | 3 | 11 79 81 |
| 159 | 3 | 11 83 85 |
| 160 | 4 | 78 81 90 92 |
| 161 | 4 | 83 85 86 88 |
| 162 | 4 | 83 84 87 88 |
| 163 | 4 | 83 84 86 89 |
| 164 | 4 | 82 85 87 88 |
| 165 | 4 | 82 85 86 89 |
| 166 | 4 | 82 84 87 89 |
| 167 | 4 | 78 80 91 92 |
| 168 | 4 | 78 80 90 93 |
| 169 | 4 | 22-23-25 38 |
| 170 | 4 | 82 84 90 92 |
| 171 | 4 | 13 20-22-25 |
| 172 | 4 | 78 81 86 88 |
| 173 | 4 | 78 80 87 88 |
| 174 | 4 | 78 80 90 92 |
| 175 | 4 | 78 80 86 89 |
| 176 | 4 | 20-22-25 27 |
| 177 | 4 | 83 84 86 88 |
| 178 | 4 | 82 85 86 88 |
| 179 | 4 | 82 84 87 88 |
| 180 | 4 | 82 84 86 89 |
| 181 | 4 | 20-23-25 27 |
| 182 | 4 | 79 81 86 88 |
| 183 | 4 | 79 80 87 88 |
| 184 | 4 | 79 81 90 92 |
| 185 | 4 | 79 80 91 92 |
| 186 | 4 | 79 80 90 93 |
| 187 | 4 | 78 81 91 92 |
| 188 | 4 | 78 81 90 93 |
| 189 | 4 | 78 80 91 93 |
| 190 | 4 | 78 80 86 88 |
| 191 | 4 | 79 80 86 89 |
| 192 | 4 | 78 81 87 88 |
| 193 | 4 | 79 81 87 88 |
| 194 | 4 | 79 81 86 89 |
| 195 | 4 | 79 80 87 89 |
| 196 | 4 | 78 81 87 89 |
| 197 | 4 | 13 21-22-25 |
| 198 | 4 | 13-22-25 38 |
| 199 | 4 | 13 20-23-25 |

Appendix B. Injection Function Cut Sets (continued)

TABLE - 2 CONTINUED :

| CUT SET NO. | NO. OF COMP. | IN C. S. | COMPONENTS NOS. |
|-------------|--------------|----------|-----------------|
| 200 | 4 | | 14 21-22-25 |
| 201 | 4 | | 14-22-25 38 |
| 202 | 4 | | 14 20-23-25 |
| 203 | 4 | | 21-22-25 27 |
| 204 | 4 | | -22-25 27 38 |
| 205 | 4 | | 21-22-25 26 |
| 206 | 4 | | -22-25 26 38 |
| 207 | 4 | | 78 81 86 89 |
| 208 | 4 | | 78 80 87 89 |
| 209 | 4 | | 14 20-22-25 |
| 210 | 4 | | 20 22-23-25 |
| 211 | 4 | | 83 84 90 92 |
| 212 | 4 | | 82 84 86 88 |
| 213 | 4 | | 82 85 90 92 |
| 214 | 4 | | 83 85 90 92 |
| 215 | 4 | | 83 84 91 92 |
| 216 | 4 | | 83 84 90 93 |
| 217 | 4 | | 82 85 91 92 |
| 218 | 4 | | 82 85 90 93 |
| 219 | 4 | | 82 84 91 93 |
| 220 | 4 | | 82 84 91 92 |
| 221 | 4 | | 82 84 90 93 |
| 222 | 4 | | 20-22-25 26 |
| 223 | 4 | | 83 85 87 88 |
| 224 | 4 | | 83 85 86 89 |
| 225 | 4 | | 83 84 87 89 |
| 226 | 4 | | 82 85 87 89 |
| 227 | 4 | | 79 80 86 88 |
| 228 | 4 | | 20-23-25 26 |
| 229 | 4 | | 79 81 91 92 |
| 230 | 4 | | 79 81 90 93 |
| 231 | 4 | | 79 80 91 93 |
| 232 | 4 | | 78 81 91 93 |
| 233 | 4 | | 21 22-23-25 |
| 234 | 4 | | 79 81 87 89 |
| 235 | 4 | | 13 21-23-25 |
| 236 | 4 | | 13-23-25 38 |
| 237 | 4 | | 14 21-23-25 |
| 238 | 4 | | 14-23-25 38 |
| 239 | 4 | | 21-23-25 27 |
| 240 | 4 | | -23-25 27 38 |
| 241 | 4 | | 21-23-25 26 |
| 242 | 4 | | -23-25 26 38 |
| 243 | 4 | | 83 85 91 92 |
| 244 | 4 | | 83 85 90 93 |
| 245 | 4 | | 83 84 91 93 |
| 246 | 4 | | 82 85 91 93 |
| 247 | 4 | | 79 80 90 92 |
| 248 | 4 | | 83 85 87 89 |
| 249 | 4 | | 79 81 91 93 |
| 250 | 4 | | 83 85 91 93 |

Appendix B. Injection Function Cut Sets (continued)

| | | | | | | | |
|---|-----------|--------|---------|------|--|-----|-----|
| 1 | 1NNSAFTY | 1.E-20 | | 30. | | 20. | .10 |
| 1 | 2 | MAINT | | | | | |
| 2 | 1125VDCD5 | | .000037 | | | | |
| 2 | 2 | | | | | | |
| 3 | 1125VDCD8 | | .000037 | | | | |
| 3 | 2 | | | | | | |
| 4 | 1250VDCD9 | | .000037 | | | | |
| 4 | 2 | | | | | | |
| 5 | 1INIT TST | | | | | | |
| 5 | 2SYS DOWN | | | | | | |
| 6 | 1LVL CCF | .43 | .001 | 90. | | 0.5 | |
| 6 | 2REACTOR | | | | | | |
| 7 | 1HW HI PR | .15 | .001 | 90. | | 0.5 | |
| 7 | 2CCF | | | | | | |
| 8 | 1K1 NOFO | 0.4 | | 365. | | | |
| 8 | 2LL RE WA | | | | | | |
| 9 | 1K2 NOFO | 0.4 | | 365. | | | |
| 9 | 2LL RE WT | | | | | | |
| 10 | 1 K3 NOFO | 0.4 | | 365. | | | |
| 10 | 2 HI DW P | | | | | | |
|5...1 0...5...2 0...5...3 0...5...4 0...5...5 0...5... | | | | | | | |
| LEGEND: LAMBDA QRESID T2 T1 TAU TREP QOVR PTCF | | | | | | | |
|5...1 0...5...2 0...5...3 0...5...4 0...5...5 0...5... | | | | | | | |
| 11 | 1 K4 NOFO | 0.4 | | 365. | | | |
| 11 | 2 HI DW P | | | | | | |
| 12 | 1K23 NOFO | 0.4 | | 365. | | | |
| 12 | 2 SEAL IN | | | | | | |
| 13 | 1V-20F SC | .15 | | 30.0 | | 20. | |
| 13 | 2CST SUPL | | | | | | |
| 14 | 1CST ISO | .04 | | 14.0 | | 20. | |
| 14 | 2 LOFC | | | | | | |
| 15 | 1V-56 ISO | .04 | | 14.0 | | 20. | |
| 15 | 2SUP LOFC | | | | | | |
| 16 | 1V-35 SUP | 1.5 | | 30.0 | | 20. | |
| 16 | 2 NCFC | | | | | | |
| 17 | 1V-36 SUP | 1.5 | | 30.0 | | 20. | |
| 17 | 2 NCFC | | | | | | |
| 18 | 1V-39 SUP | .15 | | 365. | | 19. | |
| 18 | 2 CF-FC | | | | | | |
| 19 | 1P RP/BLK | | .000001 | | | | |
| 19 | 2SUPP SUC | | | | | | |
| 20 | 1-35 LIM | 1.3 | | 30.0 | | 20. | |
| 20 | 2 NOFO | | | | | | |
|5...1 0...5...2 0...5...3 0...5...4 0...5...5 0...5... | | | | | | | |
| LEGEND: LAMBDA QRESID T2 T1 TAU TREP QOVR PTCF | | | | | | | |
|5...1 0...5...2 0...5...3 0...5...4 0...5...5 0...5... | | | | | | | |
| 21 | 1-36 LIM | 1.3 | | 30.0 | | 20. | |
| 21 | 2 NOFO | | | | | | |
| 22 | 1-35 LIM | .02 | | 30.0 | | 4.0 | |
| 22 | 2 NOFC | | | | | | |
| 23 | 1-36 LIM | .02 | | 30.0 | | 4.0 | |
| 23 | 2 NOFC | | | | | | |
| 24 | 1K15 NCFC | 0.4 | | 30.0 | | 8.0 | |
| 24 | 2 F ENER | | | | | | |
| 25 | 1V-6 CST | 1.6 | | 30.0 | | 20. | |

Appendix C. HPCI Injection Function Components

| | | | | | | | | | |
|--|---|----------|---------------|-------|------|--|-----|-----|------------|
| 25 | 2 | NOFC | | | | | | | |
| 26 | 1 | SUP WATR | .66 | | | | | 20. | |
| 26 | 2 | FSIG HI | | | | | | | |
| 27 | 1 | K25 NOFC | .02 | | | | | 8.0 | |
| 27 | 2 | SP HI WT | | | | | | | |
| 28 | 1 | PS2390A | .33 | .001 | 30.0 | | | 1.0 | .01 |
| 28 | 2 | NCFC CST | | | | | | | |
| 29 | 1 | PS2390B | .33 | .001 | 30.0 | | | 1.0 | .01 |
| 29 | 2 | NCFC CST | | | | | | | |
| 30 | 1 | V-3 NCFC | 17. | | 30.0 | | | 24. | |
| 30 | 2 | | | | | | | | |
|5...1 0...5...2 0...5...3 0...5...4 0...5...5 0...5.. | | | | | | | | | |
| LEGEND: LAMBDA QRESID T2 T1 TAU TREP QOVR PTCF | | | | | | | | | |
|5...1 0...5...2 0...5...3 0...5...4 0...5...5 0...5.. | | | | | | | | | |
| 31 | 1 | V-4 NOFC | 1.E-20.000007 | | 21.0 | | 1.5 | 24. | 0.08 .0013 |
| 31 | 2 | | | | | | | | |
| 32 | 1 | V-5 NOFC | 1.E-20.000007 | | 21.0 | | 1.5 | 24. | .0013 |
| 32 | 2 | | | | | | | | |
| 33 | 1 | PUMPS F | 3.9 | | 30.0 | | | 20. | |
| 33 | 2 | | | | | | | | |
| 34 | 1 | TURBINE | 11.5 | | 30.0 | | | 20. | .01 |
| 34 | 2 | PARTS F | | | | | | | |
| 35 | 1 | LUB PUMP | 7.9 | | 30.0 | | | 20. | |
| 35 | 2 | | | | | | | | |
| 36 | 1 | COOLER F | 3.0 | | | | | 20. | |
| 36 | 2 | AND REQ | | | | | | | |
| 37 | 1 | HPCILOCA | | .0001 | | | | | |
| 37 | 2 | | | | | | | | |
| 38 | 1 | V-6 NOFO | 1.6 | | 30.0 | | | 20. | |
| 38 | 2 | SUP NEED | | | | | | | |
| 39 | 1 | CV-45 SC | .15 | | 30.0 | | | 20. | |
| 39 | 2 | STM EXH | | | | | | | |
| 40 | 1 | V-74LOFC | .04 | | 14.0 | | | 20. | |
| 40 | 2 | STM EXH | | | | | | | |
|5...1 0...5...2 0...5...3 0...5...4 0...5...5 0...5.. | | | | | | | | | |
| LEGEND: LAMBDA QRESID T2 T1 TAU TREP QOVR PTCF | | | | | | | | | |
|5...1 0...5...2 0...5...3 0...5...4 0...5...5 0...5.. | | | | | | | | | |
| 41 | 1 | DIS PIPE | .03 | | 30.0 | | | 48. | |
| 41 | 2 | RUPTURE | | | | | | | |
| 42 | 1 | AO-7 SC | .22 | | 365. | | | | |
| 42 | 2 | | | | | | | | |
| 43 | 1 | V-8 NCFC | 1.5 | | 30.0 | | | 20. | |
| 43 | 2 | DIS ISO | | | | | | | |
| 44 | 1 | V-9 NOFC | .15 | | 30. | | | 20. | |
| 44 | 2 | DIS INJ | | | | | | | |
| 44 | 1 | V-14 FC | 1.5 | | 30.0 | | | 20. | |
| 45 | 2 | MIN RECI | | | | | | | |
| 46 | 1 | FW ISO V | 0.0 | | | | | | |
| 46 | 2 | NOFC | | | | | | | |
| 47 | 1 | FWCV-58B | 0.0 | | | | | | |
| 47 | 2 | NOFC | | | | | | | |
| 48 | 1 | V-10NCFO | .15 | | 30. | | | 20. | |
| 48 | 2 | TEST V | | | | | | | |
| 49 | 1 | V-15NCFO | .15 | | 30.0 | | | 20. | |
| 49 | 2 | TEST V | | | | | | | |

Appendix C. HPCI Injection Function Components

```

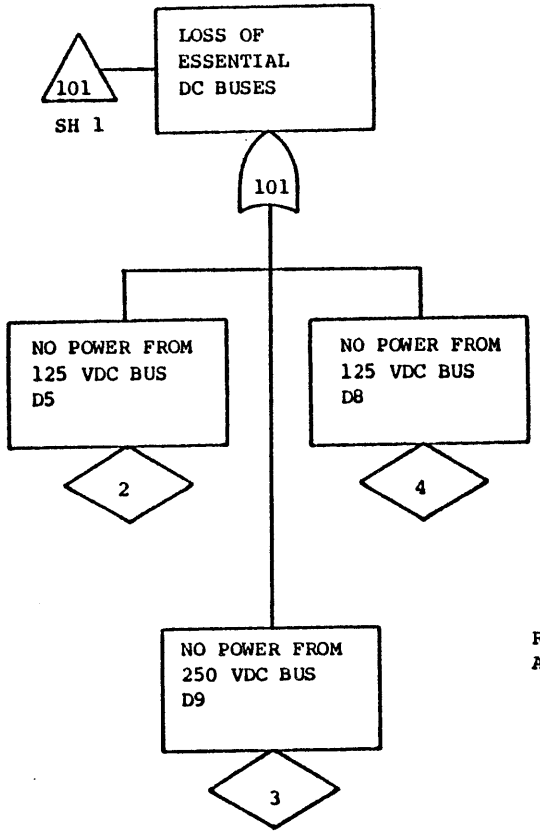
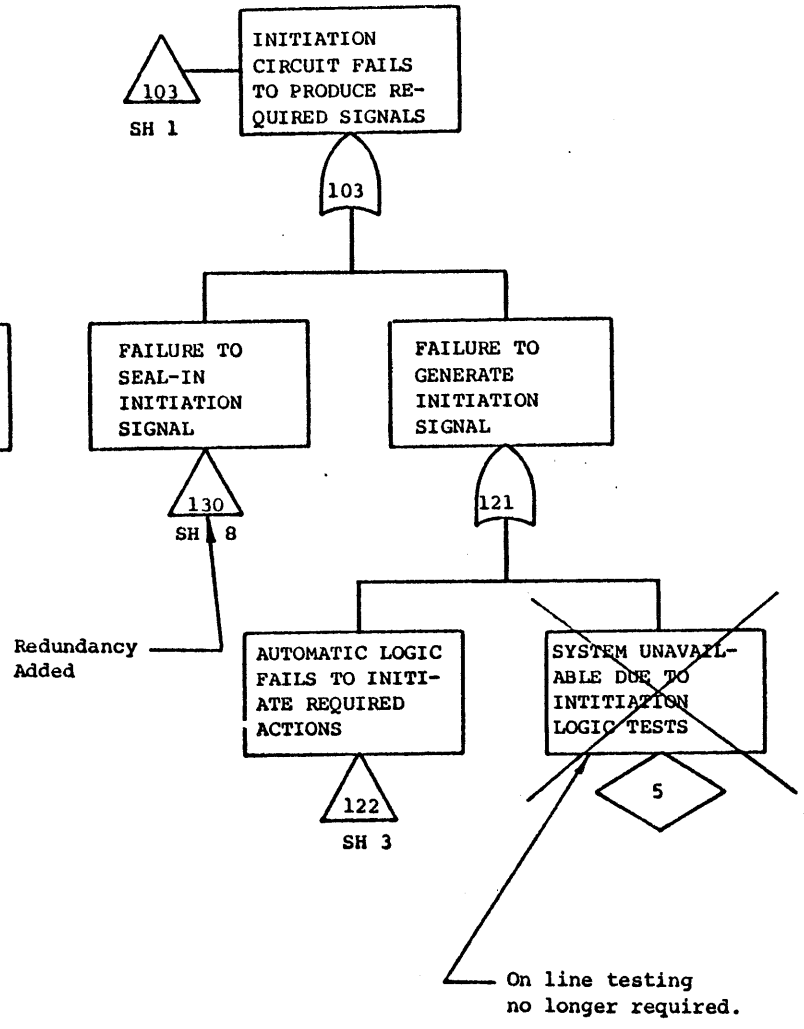
50 1RESET .0001
50 2 HEP
....5...1|0...5...2|0...5...3|0...5...4|0...5...5|0...5...
LEGEND: LAMBDA QRESID T2 T1 TAU TREP QOVR PTCF
....5...1|0...5...2|0...5...3|0...5...4|0...5...5|0...5...
51 1LWPR CCF .0001 63. 21.
51 2 F SIG
52 1TS CCF .0001 63.
52 2
53. 1TRIP CCF 0.0
53 2
54 1K28 NOFC .02 8.0
54 2 ISO REL
55 1TUR OVSP 1.0 30.0 24.0
55 2 F SIG
56 1PSL 2360 .02 365. 4.0
56 2 SHORTS
57 1K17 NOFC .02 8.0
57 2 SUC LPR
58 1TS 2371A .33 63.
58 2A T/P RM
59 1TS 2373A .33 63.
59 2A T/P RM
60 1TS 2371B .33 63.
60 2B T/P RM
....5...1|0...5...2|0...5...3|0...5...4|0...5...5|0...5...
LEGEND: LAMBDA QRESID T2 T1 TAU TREP QOVR PTCF
....5...1|0...5...2|0...5...3|0...5...4|0...5...5|0...5...
61 1TS 2373B .33 63.
61 2B T/P RM
62 1DP2352/3 .33 63. 21.
62 2ANY OF 4
63 1K9/K36 .04 8.0
63 2NOFC DP
64 1TS 2370D .33 63.
64 2B VLV ST
65 1TS 2372D .33 63.
65 2B VLV ST
66 1TS 2371D .33 63.
66 2B TORUS
67 1TS 2373D .33 63.
67 2F CLOSED
68 1TS 2370C .33 63.
68 2A VLV ST
69 1TS 2372C .33 63.
69 2F CLOSED
70 1PS2368AB .03 .000006 365. 1.0
70 2T X FSIG
....5...1|0...5...2|0...5...3|0...5...4|0...5...5|0...5...
LEGEND: LAMBDA QRESID T2 T1 TAU TREP QOVR PTCF
....5...1|0...5...2|0...5...3|0...5...4|0...5...5|0...5...
71 1K13 NOFC .02 8.0
71 2
72 1K12 NOFC .02 8.0
72 2

```

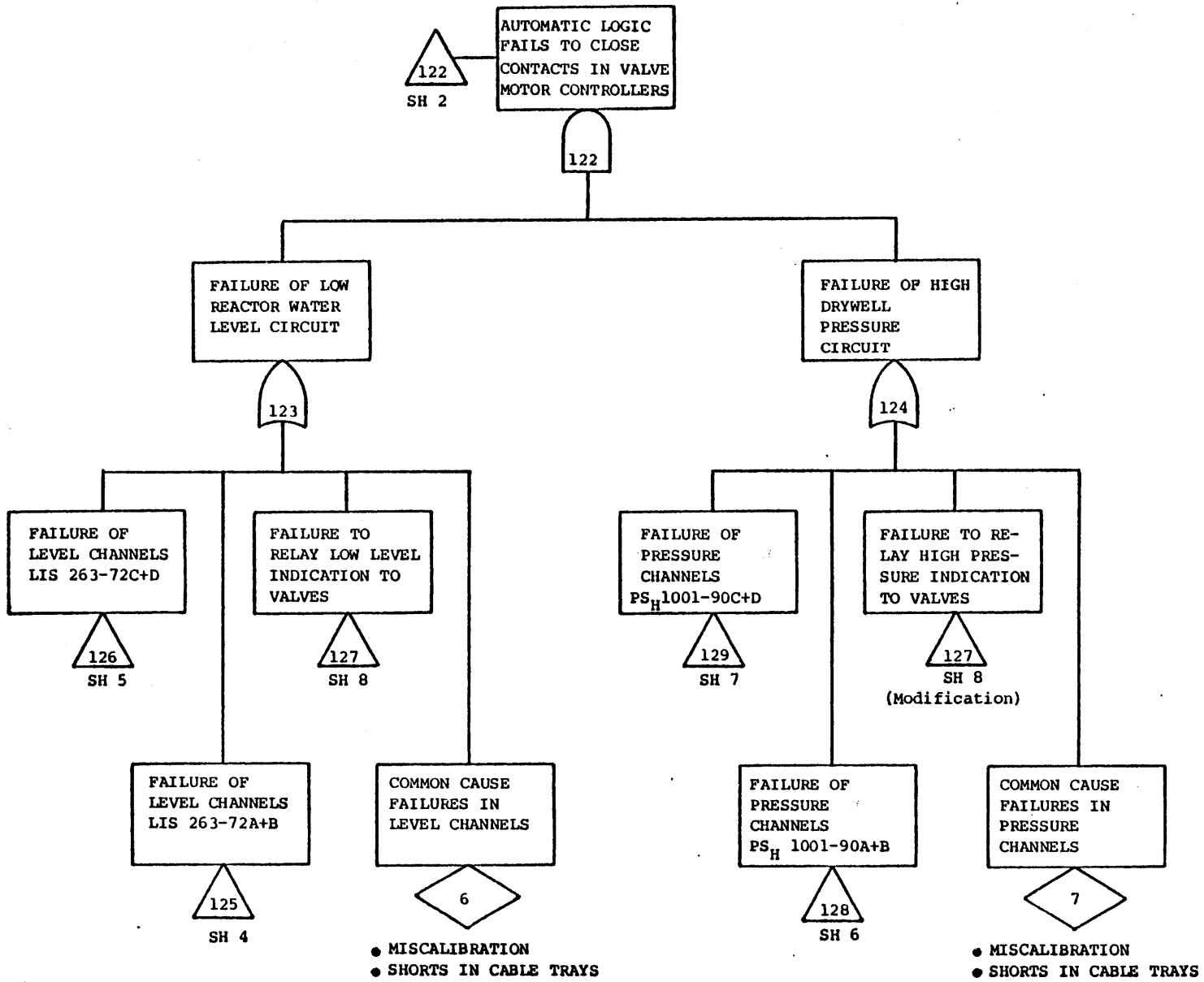
Appendix C. HPCI Injection Function Components

| | | | | | | | | | |
|----|--|------|---------|------|------|-----|-----|-----|--|
| 73 | 1PS2389AB | .66 | | 63. | 42. | | | | |
| 73 | 2STM LPFS | | | | | | | | |
| 74 | 1PS2389CD | .66 | | 63. | 42. | | | | |
| 74 | 2STM LPFS | | | | | | | | |
| 75 | 1K6/K34 | .04 | | | | | | 8.0 | |
| 75 | 2NOFC | | | | | | | | |
| 76 | 1K8/K35 | .04 | | | | | | 8.0 | |
| 76 | 2NOFC | | | | | | | | |
| 77 | 1K20 NOFC | .02 | | | | | | 8.0 | |
| 77 | 2HI T EXP | | | | | | | | |
| 78 | 1LIS-72A | 4.3 | | 30.0 | 9.0 | .25 | .25 | 1.0 | |
| 78 | 2RLV NOFO | | | | | | | | |
| 79 | 1-72A CAL | | | | | | | | |
| 79 | 2 | | | | | | | | |
| 80 | 1LIS-72B | 4.3 | | 30.0 | 9.2 | .25 | .25 | 1.0 | |
| 80 | 2RLV NOFO | | | | | | | | |
| |5....1 0....5....2 0....5....3 0....5....4 0....5....5 0....5.... | | | | | | | | |
| | LEGEND: LAMBDA QRESID T2 T1 TAU TREP QOVR PTCF | | | | | | | | |
| |5....1 0....5....2 0....5....3 0....5....4 0....5....5 0....5.... | | | | | | | | |
| 81 | 1-72B CAL | | | | | | | | |
| 81 | 2 | | | | | | | | |
| 82 | 1LIS-72C | 4.3 | | 30.0 | 9.4 | .25 | .25 | 1.0 | |
| 82 | 2RLV NOFO | | | | | | | | |
| 83 | 1-72C CAL | | | | | | | | |
| 83 | 2 | | | | | | | | |
| 84 | 1LIS-72D | 4.3 | | 30.0 | 9.6 | .25 | .25 | 1.0 | |
| 84 | 2RLV NOFO | | | | | | | | |
| 85 | 1-72A CAL | | | | | | | | |
| 85 | 2 | | | | | | | | |
| 86 | 1PS-90A | 1.5 | | 30.0 | 24. | .25 | .25 | 1.0 | |
| 86 | 2NOFO | | | | | | | | |
| 87 | 1-90A CAL | | | | | | | | |
| 87 | 2 | | | | | | | | |
| 88 | 1PS-90B | 1.5 | | 30.0 | 24.2 | .25 | .25 | 1.0 | |
| 88 | 2NOFO | | | | | | | | |
| 89 | 1-90B CAL | | | | | | | | |
| 89 | 2 | | | | | | | | |
| 90 | 1PS-90C | 1.5 | | 30.0 | 24.4 | .25 | .25 | 1.0 | |
| 90 | 2NOFO | | | | | | | | |
| |5....1 0....5....2 0....5....3 0....5....4 0....5....5 0....5.... | | | | | | | | |
| | LEGEND: LAMBDA QRESID T2 T1 TAU TREP QOVR PTCF | | | | | | | | |
| |5....1 0....5....2 0....5....3 0....5....4 0....5....5 0....5.... | | | | | | | | |
| 91 | 1-90C CAL | | | | | | | | |
| 91 | 2 | | | | | | | | |
| 92 | 1PS-90D | 1.5 | | 30.0 | 24.6 | .25 | .25 | 1.0 | |
| 92 | 2NOFO | | | | | | | | |
| 93 | 1-90D CAL | | | | | | | | |
| 93 | 2 | | | | | | | | |
| 94 | 1TS 2371C | .33 | | 63. | | | | | |
| 94 | 2A TORUS | | | | | | | | |
| 95 | 1TS 2373C | .33 | | 63. | | | | | |
| 95 | 2F CLOSED | | | | | | | | |
| 96 | 1K24REDUN | 0.30 | 3.6E-06 | 365. | | | | 8.0 | |
| 96 | 2 SEAL IN | | | | | | | | |
| | -1 | | | | | | | | |

Appendix C. HPCI Injection Function Components



Appendix D. HPCI Injection Function Fault Tree Modifications Resulting From Initiation Logic Changes (Sheet 1 of 3)



Appendix D. HPCI Injection Function Fault Tree Modifications
 Resulting From Initiation Logic Changes (Sheet 3 of 3)

318

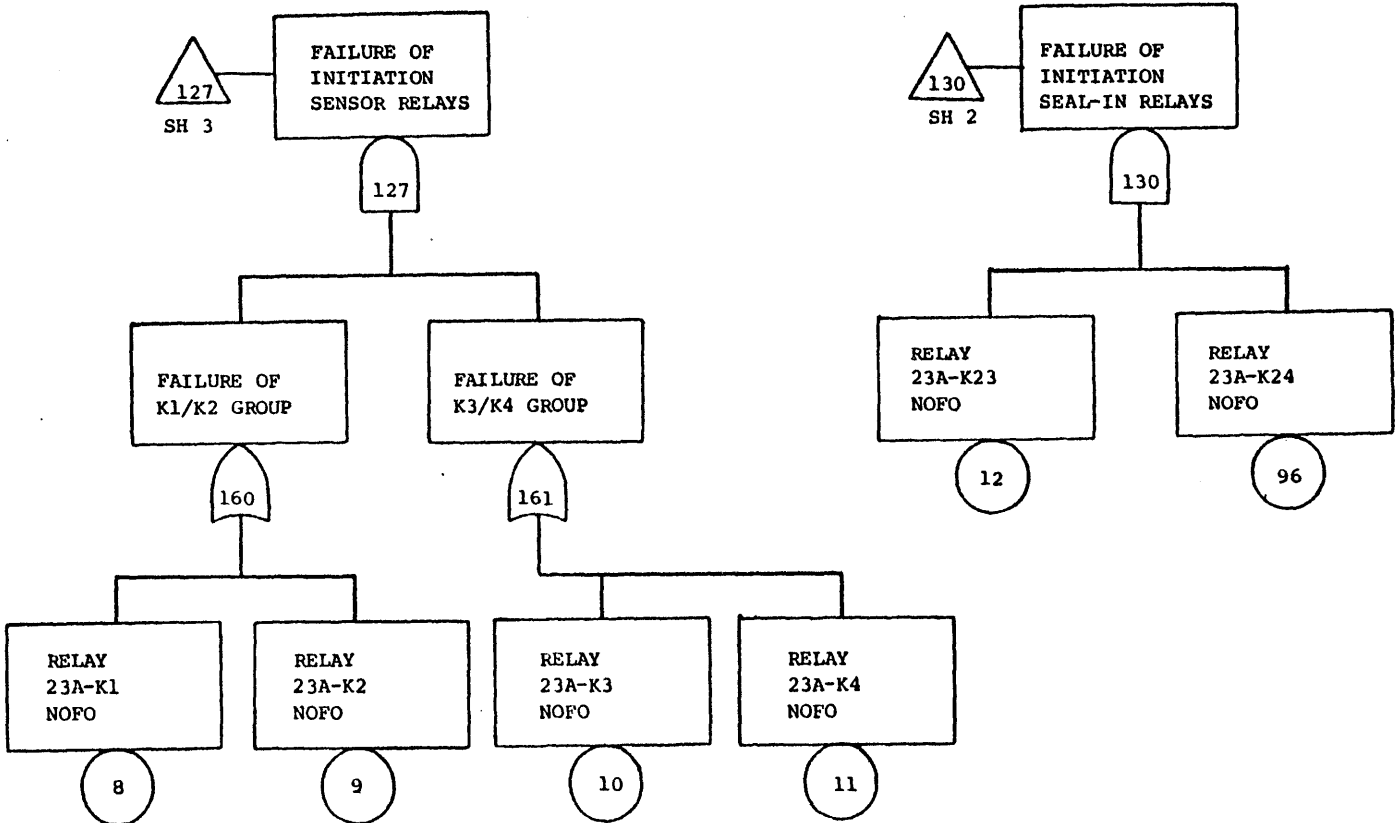


TABLE -1 FAULT TREE LOGIC

| GATE NO. | GATE TYPE | INPUT | COMP. | OR | GATES |
|----------|-----------|-------|-------|-----|-------|
| 121 | 0 | 122 | 5 | 0 | 0 |
| 122 | 1 | 123 | 124 | 0 | 0 |
| 123 | 0 | 125 | 126 | 127 | 6 |
| 124 | 0 | 128 | 129 | 130 | 7 |
| 125 | 1 | 131 | 134 | 0 | 0 |
| 126 | 1 | 133 | 134 | 0 | 0 |
| 127 | 0 | 8 | 9 | 12 | 0 |
| 128 | 1 | 137 | 138 | 0 | 0 |
| 129 | 1 | 137 | 138 | 0 | 0 |
| 130 | 0 | 10 | 11 | 12 | 0 |
| 131 | 0 | 78 | 79 | 0 | 0 |
| 132 | 0 | 80 | 81 | 0 | 0 |
| 133 | 0 | 82 | 83 | 0 | 0 |
| 134 | 0 | 84 | 85 | 0 | 0 |
| 135 | 0 | 86 | 87 | 0 | 0 |
| 136 | 0 | 88 | 89 | 0 | 0 |
| 137 | 0 | 90 | 91 | 0 | 0 |
| 138 | 0 | 92 | 93 | 0 | 0 |

Note: Gate numbers for the partial fault trees do not agree entirely with those shown in the injection function fault tree. Since gate numbers are not part of the final cut sets, they can be renumbered to suit the convenience of the analyst so long as they still convey the correct logic.

Appendix E. Initiation Function Cut Sets Before Modifications

| CUT SET NO. | NO. OF COMP. IN C. S. | COMPONENTS NOS. |
|-------------|-----------------------|-----------------|
| 1 | 1 | 5 |
| 2 | 1 | 12 |
| 3 | 2 | 8 10 |
| 4 | 2 | 7 8 |
| 5 | 2 | 9 10 |
| 6 | 2 | 6 10 |
| 7 | 2 | 6 7 |
| 8 | 2 | 9 11 |
| 9 | 2 | 8 11 |
| 10 | 2 | 7 9 |
| 11 | 2 | 6 11 |
| 12 | 3 | 10 82 84 |
| 13 | 3 | 7 82 84 |
| 14 | 3 | 8 91 92 |
| 15 | 3 | 8 90 92 |
| 16 | 3 | 9 90 92 |
| 17 | 3 | 6 90 92 |
| 18 | 3 | 8 91 93 |
| 19 | 3 | 8 90 93 |
| 20 | 3 | 6 91 92 |
| 21 | 3 | 10 78 84 |
| 22 | 3 | 9 91 92 |
| 23 | 3 | 6 90 93 |
| 24 | 3 | 11 78 84 |
| 25 | 3 | 10 79 84 |
| 26 | 3 | 10 78 85 |
| 27 | 3 | 7 79 84 |
| 28 | 3 | 7 78 85 |
| 29 | 3 | 11 82 84 |
| 30 | 3 | 10 83 84 |
| 31 | 3 | 10 82 85 |
| 32 | 3 | 7 83 84 |
| 33 | 3 | 7 82 85 |
| 34 | 3 | 9 91 93 |
| 35 | 3 | 7 83 85 |
| 36 | 3 | 7 78 84 |
| 37 | 3 | 10 83 85 |
| 38 | 3 | 11 82 85 |
| 39 | 3 | 11 83 84 |
| 40 | 3 | 11 83 85 |
| 41 | 3 | 9 90 93 |
| 42 | 3 | 6 91 93 |
| 43 | 3 | 7 79 85 |
| 44 | 3 | 11 79 84 |
| 45 | 3 | 11 78 85 |
| 46 | 3 | 11 79 85 |
| 47 | 3 | 10 79 85 |
| 48 | 4 | 82 84 90 92 |
| 49 | 4 | 83 84 90 93 |

Appendix E. Initiation Function Cut Sets Before Modifications
(continued)

TABLE - 2 CONTINUED :

| CUT SET NO. | NO. OF COMP. IN C. S. | COMPONENTS NOS. |
|-------------|-----------------------|-----------------|
| 50 | 4 | 82 84 91 92 |
| 51 | 4 | 83 84 90 92 |
| 52 | 4 | 79 84 91 92 |
| 53 | 4 | 78 85 91 92 |
| 54 | 4 | 83 85 90 92 |
| 55 | 4 | 83 85 90 93 |
| 56 | 4 | 83 85 91 93 |
| 57 | 4 | 78 84 91 93 |
| 58 | 4 | 82 85 90 92 |
| 59 | 4 | 83 85 91 92 |
| 60 | 4 | 82 85 90 93 |
| 61 | 4 | 82 84 90 93 |
| 62 | 4 | 78 84 90 92 |
| 63 | 4 | 79 84 90 92 |
| 64 | 4 | 78 84 90 93 |
| 65 | 4 | 78 85 90 92 |
| 66 | 4 | 82 85 91 93 |
| 67 | 4 | 78 84 91 92 |
| 68 | 4 | 79 85 90 92 |
| 69 | 4 | 83 84 91 93 |
| 70 | 4 | 79 85 90 93 |
| 71 | 4 | 78 85 90 93 |
| 72 | 4 | 79 84 90 93 |
| 73 | 4 | 79 84 91 93 |
| 74 | 4 | 78 85 91 93 |
| 75 | 4 | 79 85 91 92 |
| 76 | 4 | 83 84 91 92 |
| 77 | 4 | 82 85 91 92 |
| 78 | 4 | 79 85 91 93 |
| 79 | 4 | 82 84 91 93 |

TABLE -1 FAULT TREE LOGIC

| GATE NO. | GATE TYPE | INPUT | COMP. | OR | GATES |
|----------|-----------|-------|-------|-----|-------|
| 121 | 0 | 122 | 5 | 0 | 0 |
| 122 | 1 | 123 | 124 | 0 | 0 |
| 123 | 0 | 125 | 126 | 127 | 6 |
| 124 | 0 | 128 | 129 | 130 | 7 |
| 125 | 1 | 131 | 134 | 0 | 0 |
| 126 | 1 | 133 | 134 | 0 | 0 |
| 127 | 0 | 140 | 141 | 139 | 0 |
| 128 | 1 | 137 | 138 | 0 | 0 |
| 129 | 1 | 137 | 138 | 0 | 0 |
| 130 | 0 | 140 | 141 | 139 | 0 |
| 131 | 0 | 78 | 79 | 0 | 0 |
| 132 | 0 | 80 | 81 | 0 | 0 |
| 133 | 0 | 82 | 83 | 0 | 0 |
| 134 | 0 | 84 | 85 | 0 | 0 |
| 135 | 0 | 86 | 87 | 0 | 0 |
| 136 | 0 | 88 | 89 | 0 | 0 |
| 137 | 0 | 90 | 91 | 0 | 0 |
| 138 | 0 | 92 | 93 | 0 | 0 |
| 139 | 1 | 12 | 96 | 0 | 0 |
| 140 | 1 | 8 | 10 | 0 | 0 |
| 141 | 1 | 9 | 11 | 0 | 0 |

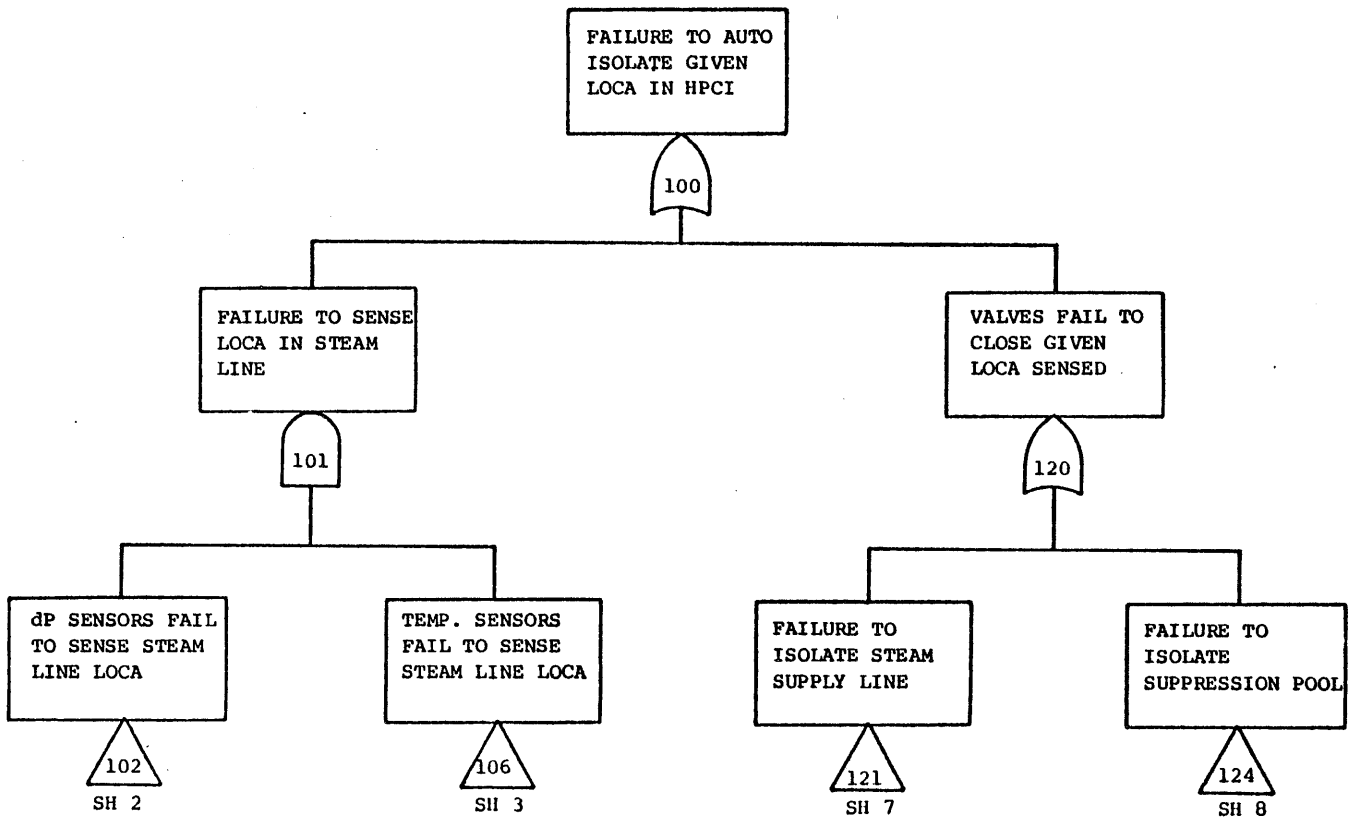
Appendix E. Initiation Function Cut Sets After Modification
(continued)

TABLE - 2 :

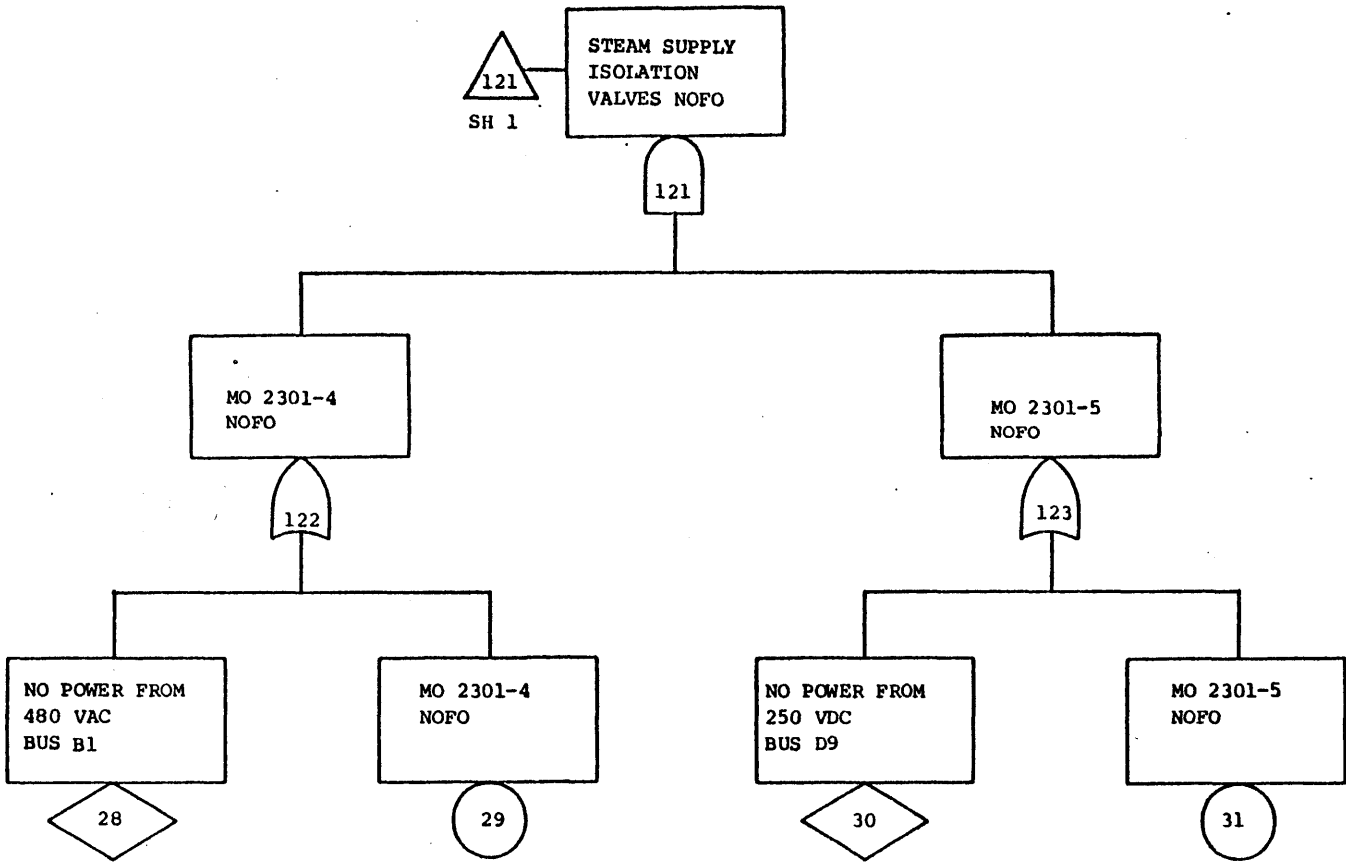
| CUT SET NO. | NO. OF COMP. IN C. S. | COMPONENTS NOS. |
|-------------|-----------------------|-----------------|
| 1 | 1 | 5 |
| 2 | 2 | 9 11 |
| 3 | 2 | 12 96 |
| 4 | 2 | 8 10 |
| 5 | 2 | 6 7 |
| 6 | 3 | 7 78 84 |
| 7 | 3 | 7 82 84 |
| 8 | 3 | 6 91 92 |
| 9 | 3 | 6 91 93 |
| 10 | 3 | 6 90 92 |
| 11 | 3 | 6 90 93 |
| 12 | 3 | 7 79 84 |
| 13 | 3 | 7 78 85 |
| 14 | 3 | 7 79 85 |
| 15 | 3 | 7 82 85 |
| 16 | 3 | 7 83 84 |
| 17 | 3 | 7 83 85 |
| 18 | 4 | 82 84 90 92 |
| 19 | 4 | 82 85 91 92 |
| 20 | 4 | 82 85 91 93 |
| 21 | 4 | 82 84 90 93 |
| 22 | 4 | 83 84 90 93 |
| 23 | 4 | 83 84 91 92 |
| 24 | 4 | 83 84 91 93 |
| 25 | 4 | 83 85 90 92 |
| 26 | 4 | 83 85 90 93 |
| 27 | 4 | 78 84 91 92 |
| 28 | 4 | 79 84 90 92 |
| 29 | 4 | 78 85 90 92 |
| 30 | 4 | 83 85 91 92 |
| 31 | 4 | 78 85 90 93 |
| 32 | 4 | 79 84 91 92 |
| 33 | 4 | 78 85 91 92 |
| 34 | 4 | 78 84 91 93 |
| 35 | 4 | 78 85 91 93 |
| 36 | 4 | 79 84 91 93 |
| 37 | 4 | 79 85 90 92 |
| 38 | 4 | 79 85 90 93 |
| 39 | 4 | 78 84 90 93 |
| 40 | 4 | 82 84 91 92 |
| 41 | 4 | 79 85 91 92 |
| 42 | 4 | 79 85 91 93 |
| 43 | 4 | 79 84 90 93 |
| 44 | 4 | 82 85 90 92 |
| 45 | 4 | 83 85 91 93 |
| 46 | 4 | 82 85 90 93 |
| 47 | 4 | 82 84 91 93 |
| 48 | 4 | 83 84 90 92 |
| 49 | 4 | 78 84 90 92 |

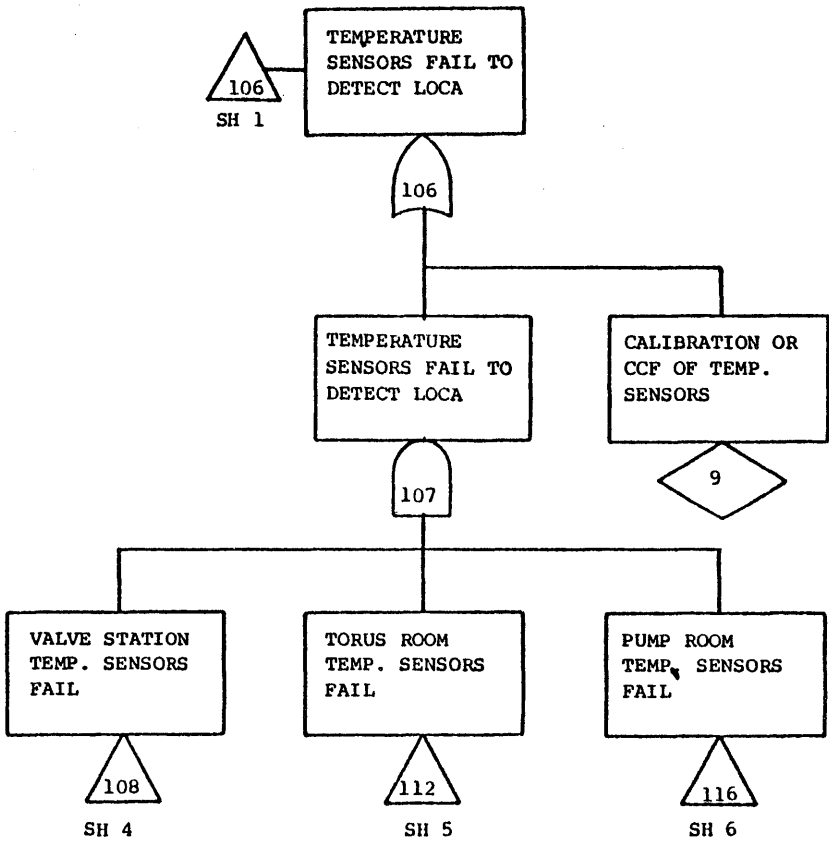
Appendix F. HPCI Autoisolation Function Fault Tree

324



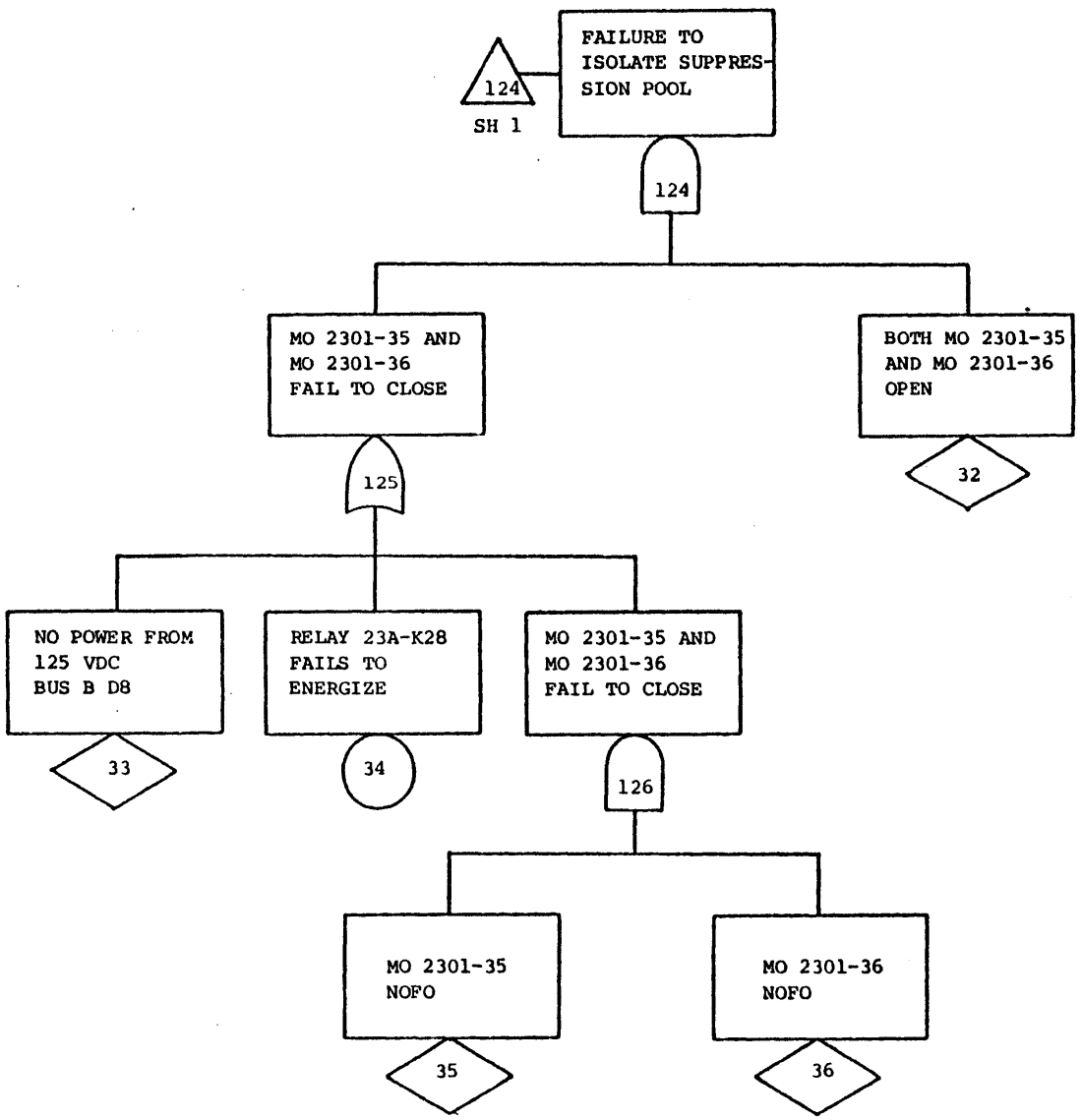
Appendix F. HPCI Autoisolation Function Fault Tree
(continued)



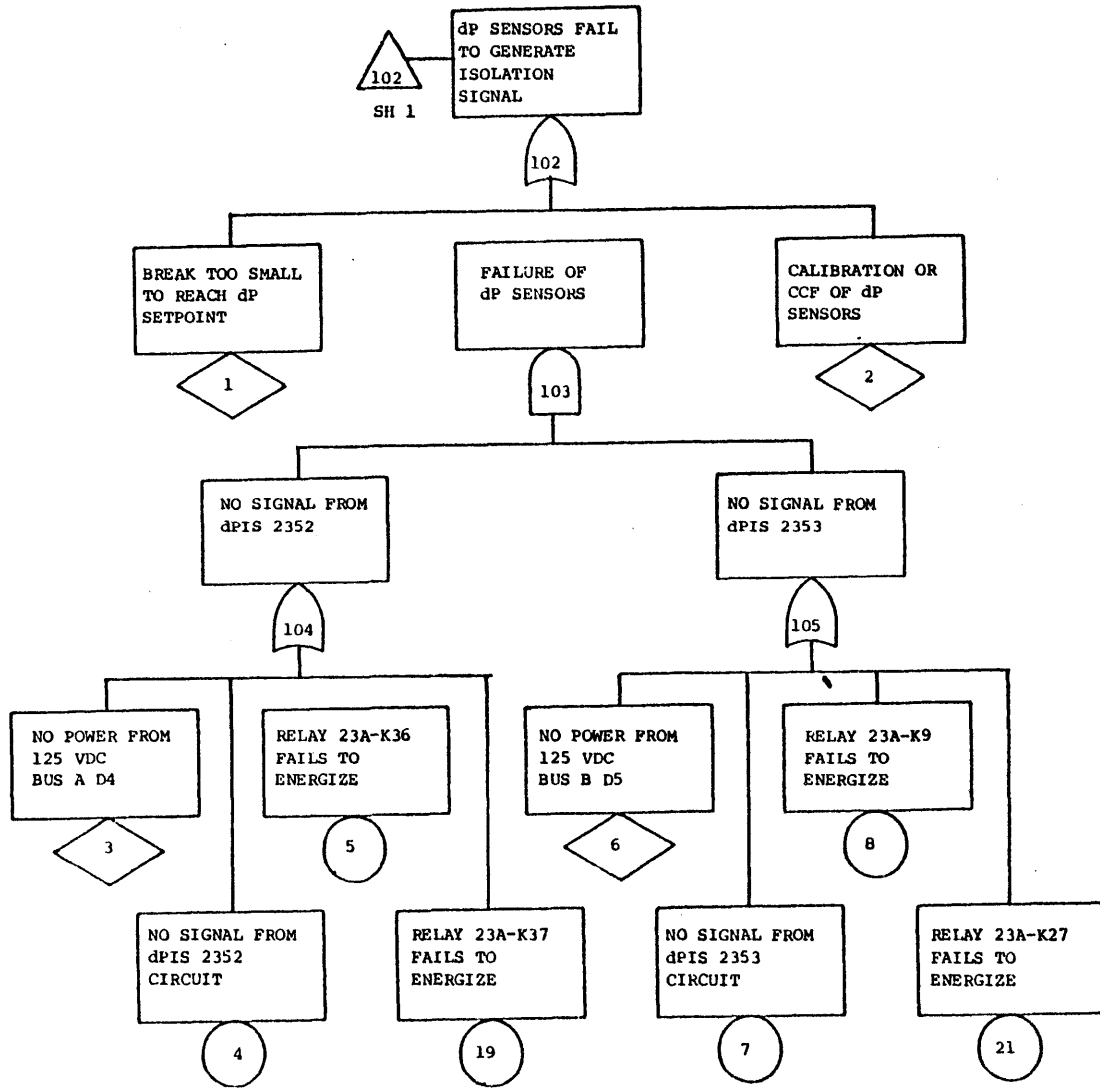


Appendix F. HPCI Autoisolation Function Fault Tree
(continued)

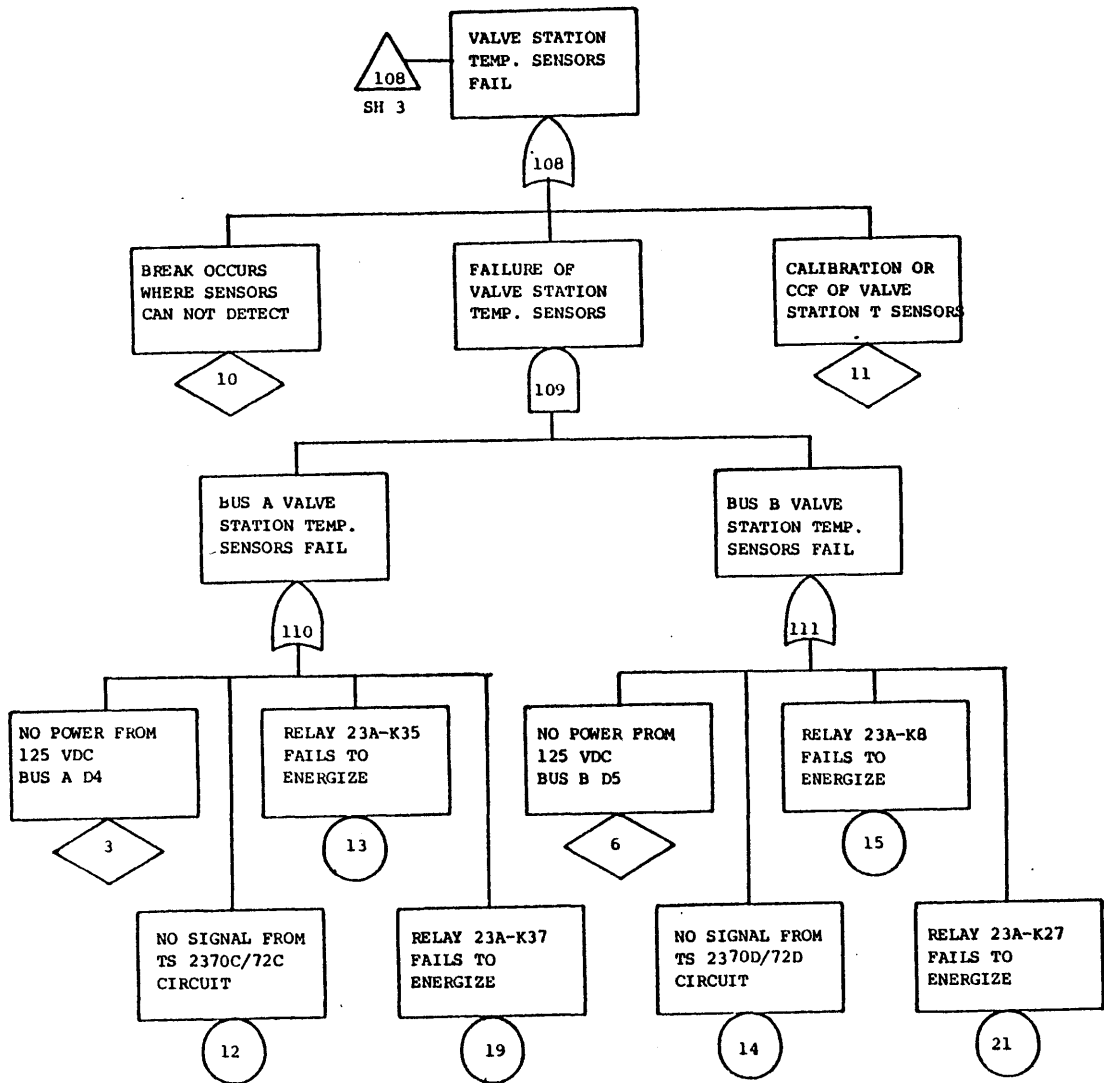
Appendix F. HPCI Autoisolation Function Fault Tree
 (continued)



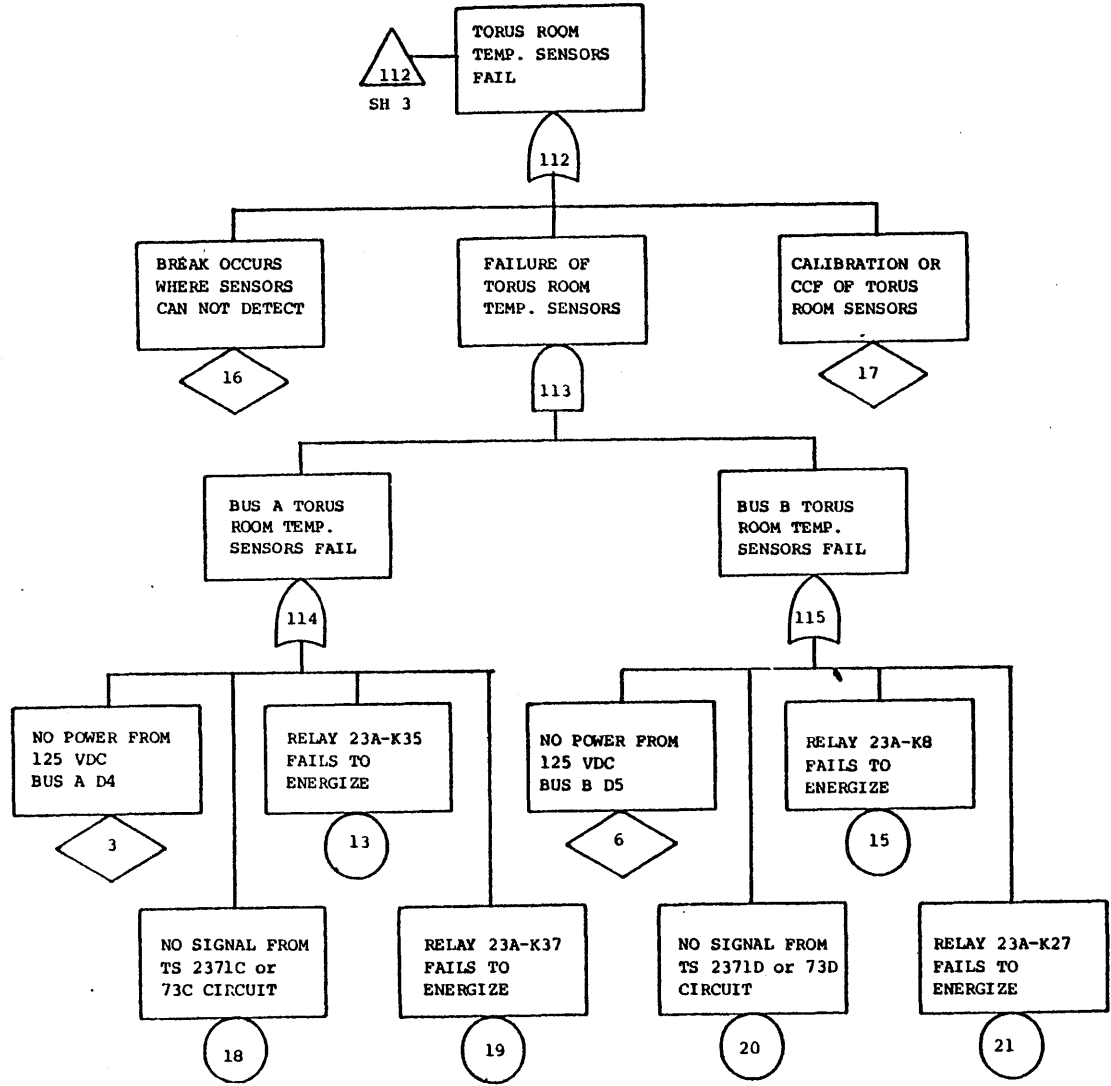
Appendix F. HPCI Autoisolation Function Fault Tree
(continued)



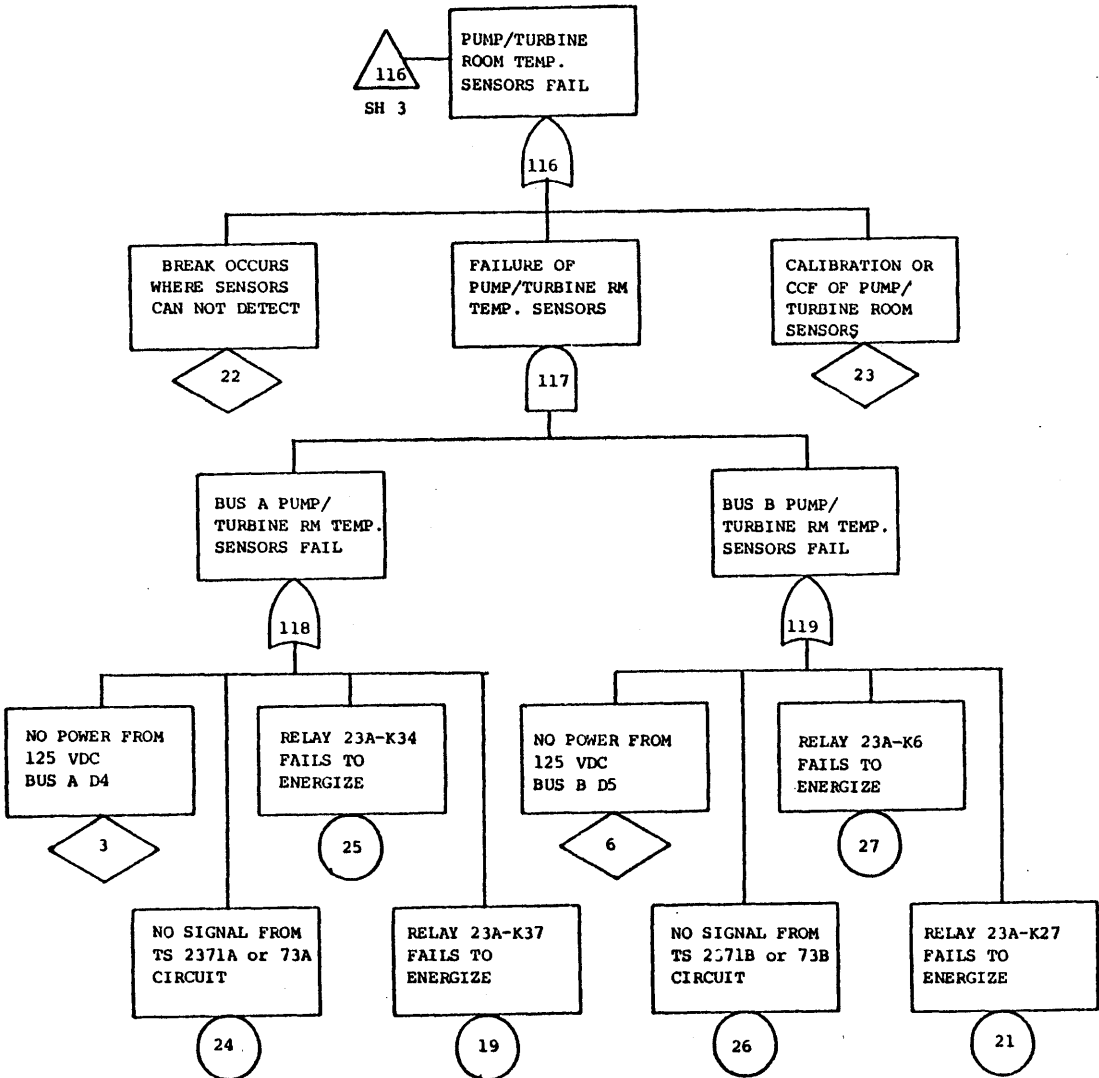
Appendix F. HPCI Autoisolation Function Fault Tree
 (continued)



Appendix F. HPCI Autoisolation Function Fault Tree
(continued)



Appendix F. HPCI Autoisolation Function Fault Tree
 (continued)



Fault Tree Logic

| GATE NO. | GATE TYPE | INPUT | COMP. | OR | GATES |
|----------|-----------|---------|-------|----|-------|
| 100 | 0 | 101 120 | 0 | 0 | 0 |
| 101 | 1 | 102 106 | 0 | 0 | 0 |
| 102 | 0 | 103 1 | 2 | 0 | 0 |
| 103 | 1 | 104 105 | 0 | 0 | 0 |
| 104 | 0 | 3 4 | 5 | 19 | 0 |
| 105 | 0 | 6 7 | 8 | 21 | 0 |
| 106 | 0 | 107 9 | 0 | 0 | 0 |
| 107 | 1 | 108 112 | 116 | 0 | 0 |
| 108 | 0 | 10 109 | 11 | 0 | 0 |
| 109 | 1 | 110 111 | 0 | 0 | 0 |
| 110 | 0 | 12 3 | 13 | 19 | 0 |
| 111 | 0 | 14 6 | 15 | 21 | 0 |
| 112 | 0 | 16 113 | 17 | 0 | 0 |
| 113 | 1 | 114 115 | 0 | 0 | 0 |
| 114 | 0 | 3 18 | 19 | 13 | 0 |
| 115 | 0 | 6 20 | 21 | 15 | 0 |
| 116 | 0 | 22 117 | 23 | 0 | 0 |
| 117 | 1 | 118 119 | 0 | 0 | 0 |
| 118 | 0 | 3 24 | 25 | 19 | 0 |
| 119 | 0 | 6 26 | 27 | 21 | 0 |
| 120 | 0 | 121 124 | 0 | 0 | 0 |
| 121 | 1 | 122 123 | 0 | 0 | 0 |
| 122 | 0 | 30 31 | 0 | 0 | 0 |
| 123 | 0 | 28 29 | 0 | 0 | 0 |
| 124 | 1 | 125 32 | 0 | 0 | 0 |
| 125 | 0 | 34 33 | 126 | 0 | 0 |
| 126 | 1 | 35 36 | 0 | 0 | 0 |

Appendix G. Autoisolation Function Cut Sets

Autoisolation Function Cut Sets

* TOTAL NUMBER OF CUT SET GENERATED = 109 *

| CUT SET NO. | NO. OF COMP. IN C. S. | COMPONENTS NOS. |
|-------------|-----------------------|-----------------|
| 1 | 2 | 28 30 |
| 2 | 2 | 32 33 |
| 3 | 2 | 6 19 |
| 4 | 2 | 29 31 |
| 5 | 2 | 32 34 |
| 6 | 2 | 28 31 |
| 7 | 2 | 29 30 |
| 8 | 2 | 1 9 |
| 9 | 2 | 2 9 |
| 10 | 2 | 3 6 |
| 11 | 2 | 3 21 |
| 12 | 2 | 19 21 |
| 13 | 3 | 32 35 36 |
| 14 | 3 | 4 7 9 |
| 15 | 3 | 4 6 9 |
| 16 | 3 | 7 9 19 |
| 17 | 3 | 5 6 9 |
| 18 | 3 | 3 7 9 |
| 19 | 3 | 5 9 21 |
| 20 | 3 | 8 9 19 |
| 21 | 3 | 3 8 9 |
| 22 | 3 | 5 8 9 |
| 23 | 3 | 4 9 21 |
| 24 | 3 | 4 8 9 |
| 25 | 3 | 5 7 9 |
| 26 | 4 | 2 13 15 22 |
| 27 | 4 | 1 13 21 23 |
| 28 | 4 | 2 11 16 22 |
| 29 | 4 | 3 7 15 23 |
| 30 | 4 | 2 10 17 22 |
| 31 | 4 | 2 10 17 23 |
| 32 | 4 | 2 10 16 23 |
| 33 | 4 | 5 13 21 23 |
| 34 | 4 | 1 13 21 22 |
| 35 | 4 | 7 15 19 22 |
| 36 | 4 | 2 10 16 22 |
| 37 | 4 | 3 8 15 22 |
| 38 | 4 | 2 11 17 22 |
| 39 | 4 | 4 6 13 23 |
| 40 | 4 | 5 6 13 24 |
| 41 | 4 | 1 11 16 23 |
| 42 | 4 | 1 11 17 23 |
| 43 | 4 | 1 13 15 23 |
| 44 | 4 | 1 6 13 23 |
| 45 | 4 | 3 7 15 26 |
| 46 | 4 | 8 15 19 26 |
| 47 | 4 | 3 7 15 27 |
| 48 | 4 | 7 15 19 23 |
| 49 | 4 | 5 13 21 22 |

| CUT SET NO. | NO. OF COMP. IN C. S. | COMPONENTS NOS. |
|-------------|-----------------------|-----------------|
| 50 | 4 | 4 6 13 25 |
| 51 | 4 | 2 6 13 24 |
| 52 | 4 | 1 6 13 22 |
| 53 | 4 | 1 13 15 22 |
| 54 | 4 | 1 11 17 22 |
| 55 | 4 | 1 13 21 25 |
| 56 | 4 | 5 6 13 22 |
| 57 | 4 | 3 7 15 22 |
| 58 | 4 | 5 6 13 25 |
| 59 | 4 | 1 10 16 22 |
| 60 | 4 | 2 3 15 22 |
| 61 | 4 | 2 11 17 23 |
| 62 | 4 | 1 15 19 26 |
| 63 | 4 | 1 11 16 22 |
| 64 | 4 | 5 13 21 24 |
| 65 | 4 | 2 6 13 22 |
| 66 | 4 | 2 6 13 23 |
| 67 | 4 | 2 13 15 23 |
| 68 | 4 | 1 10 17 22 |
| 69 | 4 | 5 13 21 25 |
| 70 | 4 | 8 15 19 27 |
| 71 | 4 | 2 6 13 25 |
| 72 | 4 | 4 6 13 24 |
| 73 | 4 | 1 15 19 22 |
| 74 | 4 | 1 6 13 24 |
| 75 | 4 | 2 13 21 22 |
| 76 | 4 | 1 6 13 25 |
| 77 | 4 | 8 15 19 23 |
| 78 | 4 | 2 13 21 25 |
| 79 | 4 | 2 13 21 24 |
| 80 | 4 | 1 15 19 27 |
| 81 | 4 | 5 6 13 23 |
| 82 | 4 | 2 13 21 23 |
| 83 | 4 | 4 6 13 22 |
| 84 | 4 | 1 10 17 23 |
| 85 | 4 | 8 15 19 22 |
| 86 | 4 | 1 15 19 23 |
| 87 | 4 | 1 13 21 24 |
| 88 | 4 | 2 3 15 27 |
| 89 | 4 | 2 3 15 23 |
| 90 | 4 | 2 15 19 22 |
| 91 | 4 | 1 3 15 22 |
| 92 | 4 | 1 3 15 23 |
| 93 | 4 | 1 3 15 27 |
| 94 | 4 | 1 3 15 26 |
| 95 | 4 | 2 15 19 23 |
| 96 | 4 | 4 13 21 22 |
| 97 | 4 | 2 11 16 23 |
| 98 | 4 | 3 8 15 27 |
| 99 | 4 | 3 8 15 23 |
| 100 | 4 | 3 8 15 26 |
| 101 | 4 | 1 10 16 23 |
| 102 | 4 | 4 13 21 23 |
| 103 | 4 | 7 15 19 27 |
| 104 | 4 | 7 15 19 26 |
| 105 | 4 | 4 13 21 25 |
| 106 | 4 | 4 13 21 24 |
| 107 | 4 | 2 3 15 26 |
| 108 | 4 | 2 15 19 27 |
| 109 | 4 | 2 15 19 26 |

Appendix G. Autoisolation Function Cut Sets
(continued)

Autoisolation Function Components

| COMP | | | | | | | | | |
|------------|-----------|-----------|-----------|-----------|---------|------|------|------|--|
| NEW | | | | | | | | | |
|5...1 | 0...5...2 | 0...5...3 | 0...5...4 | 0...5...5 | 0...5.. | | | | |
| LEGEND: | LAMBDA | QRESID | T2 | T1 | TAU | TREP | QOVR | PTCF | |
|5...1 | 0...5...2 | 0...5...3 | 0...5...4 | 0...5...5 | 0...5.. | | | | |
| 1 | 1DP CANT | .01 | | | | | | | |
| 1 | 2DETECT | | | | | | | | |
| 2 | 1CCF DP | .00010 | | | | 4.0 | | | |
| 2 | 2 | | | | | | | | |
| 3 | 1125VDCD4 | .000001 | | | | | | | |
| 3 | 2BUS A | | | | | | | | |
| 4 | 1DPIS2352 | 4.3 | .0001 | 42. | 14. | | .50 | | |
| 4 | 2NOFO | | | | | | | | |
| 5 | 1K36 NOFO | 0.3 | .00004 | 42. | 14. | | 4.0 | | |
| 5 | 2 | | | | | | | | |
| 6 | 1125VDCD5 | .000001 | | | | | | | |
| 6 | 2BUS B | | | | | | | | |
| 7 | 1DPIS2353 | 4.3 | .0001 | 42. | 14. | | .50 | | |
| 7 | 2 | | | | | | | | |
| 8 | 1K9 NOFO | 0.3 | .00004 | 42. | 14. | | 4.0 | | |
| 8 | 2 | | | | | | | | |
| 9 | 1CCF TEMP | .0001 | 42.0 | | | | | | |
| 9 | 2 ALL | | | | | | | | |
| 10 | 1VLV CANT | .01 | | | | | | | |
| 10 | 2DETECT | | | | | | | | |
|5...1 | 0...5...2 | 0...5...3 | 0...5...4 | 0...5...5 | 0...5.. | | | | |
| LEGEND: | LAMBDA | QRESID | T2 | T1 | TAU | TREP | QOVR | PTCF | |
|5...1 | 0...5...2 | 0...5...3 | 0...5...4 | 0...5...5 | 0...5.. | | | | |
| 11 | 1CCF VLV | .00010 | 42.0 | | | 8.0 | | | |
| 11 | 2TEMP SEN | | | | | | | | |
| 12 | 1TS70C72C | 6.6 | .0002 | 42.0 | | | 1.0 | | |
| 12 | 2 | | | | | | | | |
| 13 | 1K35 NOFO | 0.3 | .00004 | 42.0 | | | 4.0 | | |
| 13 | 2VLV TEMP | | | | | | | | |
| 14 | 1TS70D72D | 6.6 | .0002 | 42.0 | | | 1.0 | | |
| 14 | 2VLV TEMP | | | | | | | | |
| 15 | 1K8 NOFO | 0.3 | .00004 | 42.0 | | | 4.0 | | |
| 15 | 2VLV TEMP | | | | | | | | |
| 16 | 1TOR CANT | .01 | | | | | | | |
| 16 | 2DETECT | | | | | | | | |
| 17 | 1CCF TOR | .00010 | 42.0 | | | 8.0 | | | |
| 17 | 2TEMP SEN | | | | | | | | |
| 18 | 1TS71C73C | 6.6 | .0002 | 42.0 | | | 1.0 | | |
| 18 | 2TOR TEMP | | | | | | | | |
| 19 | 1K37 NOFO | 0.3 | .00004 | 42.0 | | | 4.0 | | |
| 19 | 2SEALIN A | | | | | | | | |
| 20 | 1TS71D73D | 6.6 | .0002 | 42.0 | | | 1.0 | | |
| 20 | 2TOR TEMP | | | | | | | | |
|5...1 | 0...5...2 | 0...5...3 | 0...5...4 | 0...5...5 | 0...5.. | | | | |
| LEGEND: | LAMBDA | QRESID | T2 | T1 | TAU | TREP | QOVR | PTCF | |
|5...1 | 0...5...2 | 0...5...3 | 0...5...4 | 0...5...5 | 0...5.. | | | | |
| 21 | 1K27 NOFO | 0.3 | .00004 | 42.0 | | | 4.0 | | |
| 21 | 2SEALIN B | | | | | | | | |
| 22 | 1PT CANT | .01 | | | | | | | |
| 22 | 2 | | | | | | | | |

| | | | | | | |
|---|-----------|------|---------|-------|--|-----|
| 23 | 1CCF P/T | | .00010 | 42.0 | | 8.0 |
| 23 | 2TEMP SEN | | | | | |
| 24 | 1TS71A73A | 6.6 | .0002 | 42.0 | | 1.0 |
| 24 | 2P/T RM | | | | | |
| 25 | 1K34 NOFO | 0.3 | .00004 | 42.0 | | 4.0 |
| 25 | 2T/P A | | | | | |
| 26 | 1TS71B73B | 6.6 | .0001 | 42.0 | | 1.0 |
| 26 | 2P/T RM | | | | | |
| 27 | 1K6 NOFO | 0.3 | .00004 | 42.0 | | 4.0 |
| 27 | 2P/T B | | | | | |
| 28 | 148OVAC | | .00004 | | | |
| 28 | 2 | | | | | |
| 29 | 1V-4 NOFO | 3.00 | .0005 | 14.0 | | 24. |
| 29 | 2 | | | | | |
| 30 | 125OVDCD9 | | .000001 | | | |
| 30 | 2 | | | | | |
|5...1 0...5...2 0...5...3 0...5...4 0...5...5 0...5... | | | | | | |
| LEGEND: LAMBDA QRESID T2 T1 TAU TREP QOVR PTCF | | | | | | |
|5...1 0...5...2 0...5...3 0...5...4 0...5...5 0...5... | | | | | | |
| 31 | 1V-5 NOFO | 3.00 | .0005 | 14.0 | | 24. |
| 31 | 2 | | | | | |
| 32 | 1V-34,35 | | .00001 | | | |
| 32 | 2BOTH OP | | | | | |
| 33 | 1125VDCD8 | | .000001 | | | |
| 33 | 2 | | | | | |
| 34 | 1K28 NOFO | 0.3 | .00004 | | | 4.0 |
| 34 | 2 | | | | | |
| 35 | 1V-35NOFO | 3.00 | .0005 | 30.00 | | 24. |
| 35 | 2 | | | | | |
| 36 | 1V-36NOFO | 3.00 | .0005 | 30.00 | | 24. |
| 36 | 2 | | | | | |
| -1 | | | | | | |

Appendix H. Autoisolation Function Components
(continued)

Appendix I

FRANTIC-II INPUT


I.1 INTRODUCTION

This appendix gives the input format of the FRANTIC II-MIT Code. Much of it is a verbatim copy or paraphrase of the FRANTIC manuals [Ve77, Ve81] so that the entire input structure can be available in this document.

The FRANTIC II-MIT Code calculates unavailability from a system unavailability equation provided by the CUTSETS subroutines, which are described in Appendix J. Before he can accomplish quantitative calculations the user must first generate cut sets for his system using CUTSETS and store the resultant output with his permanent files. The cut sets and the evaluation routines contained in CUTSETS substitute for the user provided SYSCOM subroutine required for the original FRANTIC II Code. The file containing the cut sets must be identified when calling the executive program to run the code. See Appendix K for examples of how this is done on the IBM VM/SP CMS computer system at MIT.

First
step

In addition to generating cut sets, the user must supply failure rate and test data for each component of the system. This data is broken into cases. A case contains all the input information necessary to accomplish a calculation of system unavailability over a specific interval of time. As a minimum, it contains:

- 
- Reference to the cut sets to be used to describe the system. (Only one set of minimal cut sets may be used for any one computer run. It is identified when the code is run.)
 - A set of components which make up the system whose unavailability is to be studied.
 - Titles, time, and print options (all optional).
 - Either a RUN or an OPTTEST data group to specify the type of calculation desired.

Data input for one case remains unchanged for all subsequent cases in a given computer run until specifically changed by the user. When multiple cases are run, only that data which differs from the previous case need be entered. Program execution terminates when no additional data groups are encountered. (Figure 3.1 gives a description of the computational flow.)

I.2 DATA GROUPS

Cases may be described by up to eight sets of data groups. Each data group consists of a keyword line and one or more additional lines of formatted input. A complete program run can be accomplished with two data groups. The others are optional. The eight data groups are described below.

I.2.1 TITLE DATA GROUP

This data group specifies the title for the case to be run. It consists of a keyword card containing the characters

"TITL" in the first 4 columns (only the first four characters need be entered for this and all other keyword cards) followed by a card containing 80 characters of text to be printed as a header on the output report for the case.

Input Format:

| <u>LINE</u> | <u>COLUMNS</u> | <u>NAME</u> | <u>FORMAT</u> | <u>DESCRIPTION</u> |
|-------------|----------------|-------------|---------------|--------------------|
| Type A | 1-4 | ANAME | A4 | Keyword "TITL" |
| Type B | 1-80 | TITLE1 | 20A4 | Title for output |

Sample Input:

```
TITLE  
TEST OF OFFSET CALCULATIONS FOR TEST CAUSED WEAROUT, MARCH 12
```

I.2.2 POPT DATA GROUP (OPTIONAL)

This data group is used to suppress the formatted version of FRANTIC II-MIT output contained in the original FRANTIC II code when use is being made of the FILES subroutine to write selected input and output data to a file. By selectively suppressing formatted output, the user can reduce computing time and avoid the requirement to search through a large volume of data to find one or two numbers when accomplishing sensitivity studies.

If the POPT data group is not used, all formatted output will be generated. Once suppressed, the formatted output will not be generated until another POPT data group which changes the input options is input.

Input Format:

| <u>LINE</u> | <u>COLUMNS</u> | <u>NAME</u> | <u>FORMAT</u> | <u>DESCRIPTION</u> |
|-------------|---|-------------|---------------|------------------------------------|
| A | 1-4 | ANAME | A4 | Keyword "POPT" |
| B | On this line a zero or blank suppresses the formatted printing of the data. A 1 generates the formatted output. | | | |
| | 1 | COMP | I1 | Component Data |
| | 2 | PEAK | I1 | Peak Unavailabilities |
| | 3 | QSAV | I1 | Average System Unavailability Data |
| | 4 | TIME | I1 | Time Point Data |

Sample input which suppresses time point data and peak unavailabilities.

```
POPT  
1 1
```

I.2.3 COMPONENTS DATA GROUP

This data group describes the failure characteristics of the components which make up the system to be evaluated. It is identified by a keyword card beginning with the characters "COMP." The keyword line must be followed by a line beginning with the characters "NEW" or "UPDATE". "NEW" indicates that the components being input in the data group which follows will become the component set for the next calculation. All previously input components (if any) are deleted. "UPDATE" is used to change selected parameters in existing components or to add components to an existing group.

Only the non-blank component parameters (with the exception of offset time which must be input every time) are used in updating the failure characteristics of previously input components. After the "NEW" or "UPDATE" line, two lines must be entered for each component.

The COMPONENT data group used in FRANTIC II-MIT is entirely different from the original FRANTIC II Code, as it incorporates additional failure and test parameters and provides more space for those contained in the original code. As a result, two lines of input are now necessary to describe a component's failure characteristics. Line C contains all those parameters necessary to describe failures represented by a constant hazard rate. Line D contains those which model the generalized Weibull hazard rate and test caused wearout. Unless otherwise noted the default value is zero.

Input Format:

| <u>LINE</u> | <u>COLUMN</u> | <u>NAME</u> | <u>FORMAT</u> | <u>DESCRIPTION</u> |
|-------------|---------------|-------------|---------------|--|
| A | 1-4 | ANAME | A4 | Keyword "COMP" |
| B | 1-4 | TYPE | A4 | Option NEW or UPDATE |
| C | 1-3 | INDX | I3 | Component Number |
| | 4-5 | LINE | I2 | Input line number = 1 |
| | 6-13 | NAME | A8 | Component Name |
| | 14-20 | LAMDA | F7.0 | Detectable Standby Failure Rate x $E+6/hr$, λ_0 |
| | 21-27 | QRESID | F7.0 | Demand Failure Rate, q_d |

(x10³)

| <u>LINE</u> | <u>COLUMN</u> | <u>NAME</u> | <u>FORMAT</u> | <u>DESCRIPTION</u> |
|-------------|---------------|-------------|---------------|--|
| | 28-32 | TEST2 | F5.0 | Test Interval (days), T_2 |
| | 33-37 | TEST1 | F5.0 | First Test Interval (days), T_1 , default= T_2 |
| | 38-42 | TAU | F5.0 | Average Test Time (hours), τ |
| | 43-47 | REPAIR | F5.0 | Average Repair Time (hours), T_R |
| | 48-52 | QOVRD | F5.0 | Unavailability to Override Test, q_0 |
| | 53-57 | PTCF | F5.0 | Probability of Test Caused Failures, P_f |
| | 58-62 | CAROVR | F5.0 | Test Carryover Factor, C_f |
| | 63-67 | INEFF | F5.0 | Detection Inefficiency, p |
| | 68-74 | ULAMDA | F7.0 | Undetectable Standby Failure Rate ($\times E+6/hr$), λ_μ |
| D | 1-3 | INDX | I3 | Component Number |
| | 4-5 | LINE | I2 | Input Line Number=2 |
| | 6-13 | DUMMY | | Not used for input. Space may be used to further describe comp and appears in input file only. |
| | 14-20 | BETA | F7.0 | Shape Factor, β , default = 1.0 |
| | 21-27 | OFFSET | F7.0 | Offset time (years), t_0 |
| | 28-32 | FNLAM | F5.0 | Lambda test factor, f_λ , default = 1.0 |

| <u>LINE</u> | <u>COLUMN</u> | <u>NAME</u> | <u>FORMAT</u> | <u>DESCRIPTION</u> |
|-------------|---------------|-------------|---------------|---|
| | 33-37 | FNQR | F5.0 | Demand Test Factor, f_d , default = 1.0 |
| | 38-42 | FNQT | F5.0 | Fraction of demand affected by test, f_t , default = 1.0 |
| | 43-47 | TYPE | F5.0 | Component Renewal Type: 1 = NN, 2 = OO, 3 = ON, default = 1 |

If p is input as a non-zero value, the program will re-compute λ_o as follows:

$$\lambda_o = \lambda(1-p)$$

If p is input and λ_μ is left blank, the program will compute λ_μ as follows:

$$\lambda_\mu = \lambda p$$

The 17 parameters described allow the user to specify most types of components under a variety of testing schemes: Periodically Tested Components -- user must provide λ_o , T_2 , and optionally, T_1 , τ , T_R , q_o , P_f , p , λ_μ , q_d , β , f_λ , f_d , f_t , t_o and TYPE. The following values of TYPE indicate the case:

TYPE = 1.0 GOOD AS NEW test and repair (Case NN)

= 2.0 GOOD AS OLD test and repair (Case OO)

= 3.0 GOOD AS OLD test and GOOD AS NEW repair
(Case ON)

If more than one component in series in a system is tested as the result of a single test, then set $q_o = 0$ in all but one of the components which is affected. That one component should be given the actual value of q_o . If $q_o = 1$, this procedure is

not necessary. If, however, $q_0 \neq 1$, then not setting to zero all but one of the components results in a test contribution which is over estimated, that is, q_0 is counted more than once. Monitored Components -- λ_0 and T_R must be input. The following values of ITYPE indicate the case:

TYPE = 1.0 GOOD AS NEW repair
 = 2.0 GOOD AS OLD repair

T_2 must be zero or left blank; T_1 , q_0 , f and P_f are ignored; and β , p , λ_μ , q_d and ITYPE are optional. If τ is input, it is added to T_R .

Nonrepairable Components -- λ , q_d , T_2 , T_1 , τ and T_R must be zero or left blank; P_f , p , f and ITYPE are ignored; λ_μ must be input and β is optional. Alternatively, λ may be input instead of λ_μ . In this case, T_2 should be set to a value greater than the total time period of interest, and all parameters, except β , must be zero.

Constant Unavailability -- all parameters except q_d must be zero or left blank and a value for q_d must be input.

The last line of the components data group contains "-1" in the component number field. This indicates the end of the data for the group. The maximum number of components is 100.

Sample Input:

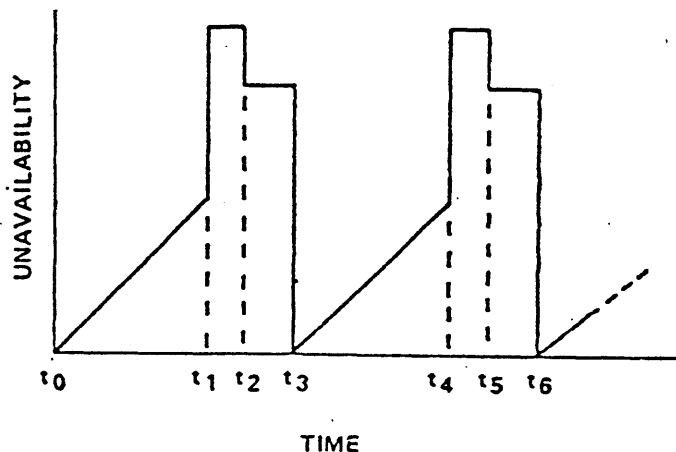
| | | | | | | |
|---|---|------|------|-----|------|---------|
| A | - | COMP | | | | |
| B | - | NEW | | | | |
| C | - | 1 | 1SUM | 20. | 30.0 | 20. .05 |
| D | - | 1 | 2 | | | |
| E | - | -1 | | | | |

Handwritten notes:
 - Arrow from "COMP" to "COMPONENT #"
 - Arrow from "NEW" to "input 1.0 or 2.0"
 - Arrow from "1" to "LAMBDA"
 - Arrow from "1" to "QRESID (?)"
 - Arrow from "2" to "LAMBDA"
 - Arrow from "-1" to "COMP #"
 - "20. x 10^4 hr^-1" near "20."
 - "What has it?" near "20. .05"

I.2.4 TIME DATA GROUP (OPTIONAL)

This data group specifies the time period over which component and system unavailabilities are to be computed. It consists of a keyword card beginning with the characters "TIME" followed by a single card containing the total time (in days) over which the time dependent, instantaneous unavailability is to be computed, and the maximum allowable interval (in days) between any two times, TDEL. If the data group (including the keyword card) is omitted, or if a zero is entered for the time period, the default value, 365 days, takes effect. If the maximum time interval, TDEL, is omitted, or if a zero is entered, no maximum will be required.

The number of time points generated by the code within the time period is a function of the test intervals, testing times, repair times of the components, as well as the value of the maximum interval. A pair of points is generated wherever a discontinuity in any component unavailability function occurs. For example, suppose a particular component has the following time dependent unavailability function:



The program will generate time points at $t_0+\epsilon$, $t_1-\epsilon$, $t_1+\epsilon$, $t_2-\epsilon$, $t_2+\epsilon$, $t_3+\epsilon$, ..., $T-\epsilon$, where ϵ is a small number ($\epsilon=10^{-4}$ hrs) and T is the total time specified in the TIME data group (or the 365 day default value). Time points for all the other components are generated in a similar manner. All the points are then merged and duplicate points discarded. A test is then made to insure that the difference between any two times is less than or equal to TDEL, and, if required, extra time points are inserted. If the system contains no periodically tested components, then time points are generated at intervals of TDEL starting at $t=0$.

The time points generated in FRANTIC II are based on all the components input in the COMPONENTS data group, regardless of whether they are actually used in the system unavailability function.

The spacing of the time points affects the accuracy of the computed average (or mean) system unavailability and the appearance of the instantaneous unavailability plots. The more non-linear the system unavailability, the more time points are required for extremely precise evaluations. A lack of sufficient points can cause some distortions in the plots and yield somewhat inaccurate estimates of the average system unavailability.

The inaccuracy occurs because the program computes the average system unavailability by numerical integration using the trapezoid rule (i.e., successive points are connected by

straight lines and the area under the resulting function is computed). The method yields the correct area for the contributions to unavailability due to testing and repairs, but may overestimate or underestimate contributions due to failures (the between tests contribution).

Input Format:

| <u>LINE</u> | <u>COLUMN</u> | <u>NAME</u> | <u>FORMAT</u> | <u>DESCRIPTION</u> |
|-------------|---------------|-------------|---------------|---|
| A | 1-4 | ANAME | A4 | Keyword "TIME" |
| B | 1-10 | TEND | F10.0 | Total Time Period In Days, T (default, T=365) |
| | 11-20 | TDEL | F10.0 | Maximum Time Interval In Days (default, no maximum) |

Sample Input:

```
TIME
1825.    5.0
```

I.2.5 PRINT DATA GROUP (OPTIONAL)

This data group is used to request a table printout of the system unavailabilities computed by the program over one or more time intervals (within the input time period) and to specify the number of instantaneous unavailabilities to be ranked by magnitude. The data group is identified by a keyword card beginning with the characters "PRIN." The keyword card must be followed by one card containing the number of time intervals desired and the number of maximum unavailabil-

ities desired. The value input for the number of intervals may be -1, 0, 1, 2, 3, or 4.

If the value input is negative, all system unavailabilities computed are printed and no additional lines are necessary. If the value is zero, any print options previously specified are nullified and the default option (no print) is instituted. No additional lines are necessary. If the value is greater than zero, another line containing the end points of the intervals is read. In this case the program will print the system unavailability at all the computed time points that fall within the specified interval(s) including the end points. A maximum of four intervals may be specified, and they may overlap.

The maximum unavailability output lists, in decreasing order, the n greatest instantaneous system unavailabilities computed by the program. The number of unavailabilities printed (n) has a default value of 12 and may not exceed 100. If the PRINT data group (including the keyword card) is omitted, 12 peaks are printed, and no other system unavailability printout is produced. The PRINT data group is depicted as follows:

Input Format:

| <u>LINE</u> | <u>COLUMN</u> | <u>NAME</u> | <u>FORMAT</u> | <u>DESCRIPTION</u> |
|--|---------------|-------------|---------------|--|
| A | 1-4 | ANAME | A4 | Keyword = "PRIN" |
| B | 1-5 | NPRT | I5 | Number of time intervals for printing system unavailabilities (-1,0, or 1-4) |
| | 6-10 | NPK | I5 | Number of peaks to be printed |
| C (use only when number of intervals is >0) | 1-10 | STPRT(1) | F10.0 | Start of first time interval in days |
| | 11-20 | FINPRT(1) | F10.0 | End of first time interval in days |
| | 21-30 | STPRT(2) | F10.0 | Start of 2nd time interval in days |
| | 31-40 | FINPRT(2) | F10.0 | End of 2nd time interval in days |
| | 41-50 | STPRT(3) | F10.0 | Start of 3rd time interval in days |
| | 51-60 | FINPRT(3) | F10.0 | End of 3rd time interval in days |
| | 61-70 | STPRT(4) | F10.0 | Start of 4th time interval in days |
| | 71-80 | FINPRT(4) | F10.0 | End of 4th time interval in days |

Sample Input:

A - PRINT
 B - 2
 C - 0.0 365. 1460. 1825.

Handwritten notes:
 - "start first interval" with arrow pointing to 365.
 - "end first interval (days)" with arrow pointing to 1460.
 - "start 2d time interval" with arrow pointing to 1825.
 - "Number of intervals for printing" written above the input line.

I.2.6 PLOT DATA GROUP (NOT ACTIVE IN FRANTIC II-MIT)

This data group specifies how instantaneous system unavailability is to be plotted. To use the plot option it would be necessary to make the plot generating subroutines in FRANTIC II-MIT conform with local plotting hardware. Since average unavailability of the system rather than its instantaneous unavailability was the primary information derived from running FRANTIC II-MIT in this study, the plot routines were not modified to function on the MIT plotter.

I.2.7 RUN DATA GROUP

This data group initiates the system unavailability computations. The COMPONENTS and optionally the TITLE, TIME, PRINT and PLOT parameters must be set up before the RUN data group. The RUN data group is identified by a keyword card beginning with the characters "RUN."

The RUN keyword card must be followed by one or more run data cards, where each card has the following parameters:

- 1) system number - number code identifying the number of terms in the inclusion-exclusion expansion used for calculations. This may be 1, 3 or 5. The use of 1 corresponds to use of the rare event approximation and is the fastest running of the 3 options.
- 2) unavailability option - four letter code selecting the type of unavailability to be computed where

"FAIL" means compute the instantaneous unavailability based on contributions

from component failures only (the between tests contribution).

"TOTL" means compute the instantaneous unavailability based on contributions from failures, testing and repairs.

When the unavailability option is left blank, the default value is "TOTL."

- 3) x-scale - four letter code specifying the scaling of the points along the x or time axis for plotting.

"NONE" means no plots are produced and should always be used in FRANTIC II-MIT.

If x-scale is left blank, the default value is "LIN" when the unavailability option is "FAIL", "BOTH" when the unavailability option is "TOTL."

- 4) y-scale - four letter code specifying the scaling of the points along the y or system unavailability axis where

"NONE" means no plots are produced (may be omitted if x-scale = "NONE")

- 5) plot cutoff option - power of 10 to be used as a lower bound on system unavailability for plotting (e.g., $-7 = 10^{-7}$). The default is no cutoff. This is not used in FRANTIC II-MIT.
- 6) plot title - 56 character text to appear as a plot subheading in addition to the title for the case.

A negative system number indicates the end of the RUN data group.

Input Format:

| <u>LINE</u> | <u>COLUMN</u> | <u>NAME</u> | <u>FORMAT</u> | <u>DESCRIPTION</u> |
|-------------|---------------|-------------|---------------|--|
| A | 1-4 | ANAME | A4 | Keyword "RUN" |
| B | 1-3 | NSYS | I3 | 1, 3 or 5 number terms in incl.-excl. expansion |
| | 5-8 | QOPT | A4 | Unavailability option |
| | 10-13 | XOPT | A4 | Input NONE |
| | 15-18 | YOPT | A4 | Input NONE or leave blank |
| | 20-23 | ICUTOP | I4 | Leave blank |
| | 25-80 | TITLE2 | 14A4 | RUN title |
| C | 1-3 | NSYS | I3 | End of run cards indicator (-1) |

Sample Input:

A - 3 RUN
 B - 1 TOTL NONE NONE
 C - -1
 ← end of run cards

RUN DATA

blank

I.2.8 TEST DATA GROUP

This data group describes the input necessary to use Subroutine OPTEST. Subroutine OPTEST calculates the optimum test interval of a single component subject to test caused down time. Test caused down time is the result of either test caused failures, unavailability to override the test, or repair of demand caused failures. The subroutine uses data input using the "COMP" data group for its calculations. Two options can be used to obtain output:

1) Calculate the optimum test interval, the (minimum) unavailability when tested at this interval, and the two test intervals which will result in an increase in the average unavailability by a factor of f over the minimum. (One will be shorter than the optimum, corresponding to an increase in test caused unavailability, and one will be longer, corresponding to an increase in undetected standby failures.)

2) Calculate the optimum test interval, the (minimum) unavailability resulting from testing at the optimum interval, the unavailability resulting from testing at the interval input in the "COMP" data group for that component, and the factor f increase of that unavailability over the minimum.

Input Format:

| <u>LINE</u> | <u>COLUMN</u> | <u>NAME</u> | <u>FORMAT</u> | <u>DESCRIPTION</u> |
|-------------|---------------|-------------|---------------|---|
| 1 A | 1-4 | ANAME | A4 | Keyword "TEST" |
| 2 0 | 1 | OPTION | I1 | 0 = Terminate OPTEST calculations 1 = Calculate T2OP, QMIN (at T2OP), T2LOW and T2HI giving $Q = F * QMIN$. 2 = Calculate T2OP, QMIN, Q for T2 input with component data, and $F = Q / QMIN$. |
| | 2-5 | INDEX | I4 | Component index number. |
| | 6-15 | F | F10.3 | Input for option 1 only. Factor increase of Q over $QMIN$ used for calculating T2LOW and T2HI. Default is 1.1. |

Sample Input:

```
A - TEST
B - 1 1
    1 1 1.2
    1 1 1.3
    1 1 1.4
    1 1 1.5
    0
```

Appendix J

THE CUTSETS PACKAGE

J.1 INTRODUCTION

Cut sets are generated by the CUTSETS package of sub-routines taken directly from Karimi [Ka80]. This appendix describes the input necessary to use CUTSETS and gives some suggestions for improving the flexibility of its use.

Procedure for using the CUTSETS subroutines:

- First derive the fault tree by an analysis of the system.
- Assign a number to each basic failure event starting from 1. For complement events, use the negative of the number of the event, (i.e., if there exist both A and A in a fault tree and the number N has already been assigned to A, then -N should be assigned to A).
- Assign a number to the Top Event.
- Number all the intermediate gates in order until all the gates have been numbered.

When assigning a number to the Top Gate it is possible to leave room for additional basic failure events. This will allow for future expansion of the fault tree without the requirement to renumber all the gates. See the usage notes.

J.2 INPUT FORMAT

| <u>LINE</u> | <u>COLUMN</u> | <u>NAME</u> | <u>FORMAT</u> | <u>DESCRIPTION</u> |
|-------------|---------------|-------------|---------------|---|
| A | 1-4 | IMAX | I4 | Largest number reserved for basic failure events in the fault tree logic. |

| <u>LINE</u> | <u>COLUMN</u> | <u>NAME</u> | <u>FORMAT</u> | <u>DESCRIPTION</u> |
|-------------|---------------|--------------|---------------|---|
| | 5-8 | IMAXT | I4 | Number of the last gate in the fault tree. <i>Total # components plus gates w fault tree</i> |
| | 9-12 | LMAX | I4 | Maximum number of inputs to a single gate. Presently, IMAX limit is 18. |
| | 13-16 | NSORT | I4 | A control word for minimal cut set print out. If NSORT=1, no sorting of cut sets. If NSORT=0, the minimum cut sets are ordered according to size. |
| | 17-20 | NMAX | I4 | Maximum allowable components per cut set. (Reduced to 10 in CUTSETS) |
| B | 1-4 | L1(I,1) | I4 | Gate number I |
| | 5-8 | L1(I,2) | I4 | Logic of Gate I: 1 = AND, 0 = OR. |
| | 9-80 | L1(I,J=3,20) | 18(I4) | Numbers of gates or components input to Gate I. Right justify within field of four spaces. Complement events are entered as negative numbers. |

The Type B input line continues until every gate has been described. The gates must be input in sequence starting from the top gate to the last gate with none left out.

When the input file is completed the first number in each line should increment consecutively from the number of components to the number of the last gate.

Sample Input:

FILE: 20OUTOF3 LOGIC A

| | | | | | |
|---|----|----|----|----|----|
| | 4 | 8 | 2 | 1 | 1 |
| A | 9 | 14 | 3 | 0 | 3 |
| C | 10 | 0 | 11 | 12 | 14 |
| | 11 | 1 | 1 | 2 | |
| | 12 | 1 | 3 | 4 | |
| | 13 | | | | |
| | 14 | 1 | 5 | 6 | |

J.3 USEAGE NOTES

1. Within a fault tree it is not important how the gates are numbered so long as an input line appears for every gate number between the Top Event and the last gate. Lower numbered gates may be input into higher numbered gates without affecting the logic of the tree.

2. If a gate is removed from the system fault tree the following procedure can be used to avoid renumbering the gates following it.

- a. Insure that the gate is removed as an input to any other gate.
- b. Leave an input line in the LOGIC file which contains only the number of the removed gate. This will satisfy the input format of the code, but the line containing the number of the removed gate will have no affect on the fault tree logic.

In the sample input file given above, the line numbered 13 represents a gate that has been removed from the fault tree logic. It can be seen in the sample output that a gate 13 is listed as part of the input, but the cutsets are generated without its affecting the results.

3. The CUTSETS routines are run on an IBM VM/SP CMS system at MIT using a file called CUTSETS EXEC:

```
GLOBAL TXTLIB FORMLIBVFORLIB PLOTLIB CMSLIB FORTLIB FORMLIB  
FILEDEF 5 DISK &1 LOGIC  
FILEDEF 6 DISK &1 CUTOUT(LRECL 132  
FILEDEF 8 DISK &1 CUTSETS(LRECL 132  
LOAD CUTSETS (START
```

In this file, &l is a variable which will assume the name of whatever file name the user desires. The first line calls the libraries necessary to interpret the FORTRAN commands. For example, the fault tree logic of the 2OUTOF3 LOGIC file can be processed by typing:

```
CUTSETS 2OUTOF3
```

The EXEC will execute in the following manner:

```
cutsets 2outof3 vFORMLIBFORTLIB
GLOBAL TXTLIB FORMLIB PLOTLIB CMSLIB FORTLIB
FILEDEF 5 DISK 2OUTOF3 LOGIC
FILEDEF 6 DISK 2OUTOF3 CUTOUT ( LRECL 132
FILEDEF 8 DISK 2OUTOF3 CUTSETS ( LRECL 132
LOAD CUTSETS ( START
EXECUTION BEGINS...
R; T=0.24/0.50 13:25:44
```

Two files are produced as a result of the program execution.

a. Formatted output for qualitative analysis is contained in 2OUTOF3 CUTOUT. This file is not used for quantitative evaluations and may be erased after printing.

```
FILE: 2OUTOF3 CUTOUT A
```

```
1          TABLE -1  FAULT TREE  LOGIC
0          GATE      GATE      INPUT COMP. OR GATES
           NO.      TYPE
           10        0         11  12  14
           11        1         1  2  0
           12        1         3  4  0
           13        0         0  0  0
           14        1         5  6  0
```

```
0 MINO IND LMIN II I
  0  0  3  4  3
```

```
1 * TOTAL NUMBER OF CUT SET GENERATED = 3 *
```

```
0          TABLE - 2 ;
          CUT SET NO.  NO. OF COMP. IN C. S.  COMPONENTS NOS.
```

```
          -----
          1          2          1  2
          2          2          3  4
          3          2          5  6
```


Appendix K

RUNNING FRANTIC II-MIT ON CMS

This appendix contains suggestions for running FRANTIC II-MIT on the IBM 370 VM/SP CMS system.

K.1 USING XEDIT COMMANDS AND MACROS

The formatted input required by FRANTIC II-MIT can be easily and accurately established using Xedit subcommands. The most useful commands are CLocate and COVerlay using the column pointer provided by the editor. If two or more commands are executed in sequence repetitively, a separate file with filetype XEDIT can be established to execute all the commands at once. For example, the following file will automatically insert a value for q_d in the current line, provided the user is on the correct line.

```
FILE: QD      XEDIT  A
```

```
CL:21  
COV &1
```

This command can be executed by typing q_d `_.0001_` and return. The symbols `_` put a blank into the file. Note that the format for inputting q_d is F7.0, so seven characters are in the field containing `.0001`. This ensures that any previous value in the space provided for q_d is completely overwritten by the current value. On a line editing CRT, the result will

look like:

```
 / 8 1
   8 1K1 NOFO 0.30 3.6E-06 60. 19.
qd .0001
  8 1K1 NOFO 0.30 3.6E-06 60. 19.
  8 1K1 NOFO 0.30 .0001 60. 19.
```

The following is an example of a file which might be used to change the periodic test interval of selected components in a system. When the command,

```
macro m2254 _30._
```

is used, this file will be used to change the periodic test interval of the specified components in the previous COMP data group. The word "macro" is required because the file name contains numbers.

```
FILE: M2254 XEDIT A
```

```
SET V OFF
L -/COMP
CL:28
L / 51 1/
COV &1
L / 73 1/
COV &1
L / 74 1/
COV &1
SET V ON
CF
```

The above file can be used to create input for sensitivity calculations of the effect of test interval on unavailability. When multiple calculations are to be made in one computer run, use can also be made of the PUT and GET subcommands for setting up the calculation. (It is not necessary to input the subcommands in capital letters. The capitals are used here to emphasize that they are subcommands of the Xedit editor.) A possible procedure is:

- 1) Set up the first calculation with the COMP and RUN data groups together at the end of the input.

2) Give the subcommand `-/COMP` to move the current line to the beginning of the COMP data group.

3) Give the subcommand `PUT*` to copy both the COMP and RUN data groups into a temporary holding file while also keeping it in the current data file.

4) Give the subcommand `BOT` to ensure the current line is at the end of the data file.

5) Give the subcommand `GET` to copy an additional COMP and RUN data group into the file. There will now be two COMP and RUN data groups in the original data file.

6) Give the subcommand `MACRO M2254 _35._` to change the test interval to 35 days. The macro will go back to the beginning of the previous COMP data group, search for the designated component identifications and overlay `_35._` in the field starting at column 28.

With the subcommands and macros available in Xedit the flexibility for setting up input data files is immense. For further information consult the IBM reference manuals.

K.2 RUN EXEC

The following is the EXEC file used to run FRANTIC II-MIT.

```
FILE: RUN      EXEC      A
GLOBAL TXTLIB ESRTMO2 VFORMLIB PLOTLIB CMSLIB FORTLIB
FILEDEF 5 DISK &3 DAT
FILEDEF 6 DISK &3 OUT(LRECL 132)
FILEDEF 8 DISK &2 CUTSETS(LRECL 132)
FILEDEF 10 DISK &3 FILE
FILEDEF 11 DISK &3 OPTST
LOAD &1 (START)
```

In order to run FRANTIC II-MIT the following files must have been established.

1) CUTSETS must have already been run to produce a file having filetype CUTSETS.

2) Input data is stored in a file with filetype DAT.

When the code runs it produces:

1) Standard formatted output in a file having the filename of the input data files name and filetype OUT.

2) Output in accordance with that specified in the FILES subroutine is stored in a file with a filename of the input data but having a filetype FILE.

The following is an example of how a run looks on a terminal. This run produced the data for Figure 5.4.

```
run frantic parallel simpsys
GLOBAL TXTLIB FORTMOD2 PLOTLIB CMLIB FORTLIB
FILEDEF 5 DISK SIMPSYS DAT
FILEDEF 6 DISK SIMPSYS OUT ( LRECL 132
FILEDEF 8 DISK PARALLEL CUTSETS ( LRECL 132
FILEDEF 10 DISK SIMPSYS FILE
FILEDEF 11 DISK SIMPSYS OPTEST
LOAD FRANTIC ( START
EXECUTION BEGINS...
R; T=1.26/1.67 00:26:47
```

K.3 SUBROUTINE FILES

The following is an example of Subroutine FILES. It is self explanatory.

FILE: FILES FORTRAN A

SUBROUTINE FILES

```
C
C SUBROUTINE FILES OUTPUTS COMMON BLOCK DATA TO A USER SPECIFIED
C FILEDEF 10. CURRENTLY THE USER MUST PROGRAM FILES FOR THE
C PARAMETERS HE DESIRES TO BE SAVED IN A FILE.
C THIS SUBROUTINE WILL BE MOST USEFUL FOR DOING SENSITIVITY STUDIES
C THE USER CAN SUPPRESS NORMAL FORMATTED OUTPUT USING THE
C POPT DATA GROUP. THE SPECIFIC VARIABLE BEING CHANGED AND THE
C RESULTING QSAVG CAN THEN BE OUTPUT TO FILEDEF 10 FOR USE IN
C PLOT ROUTINES OR SENSITIVITY ANALYSIS.
C
C
C THIS SECTION ESTABLISHES THE COMMON BLOCKS ACCESSED
COMMON /NRUN/ NRUN
COMMON /HEADER/ TITLE1(22),TITLE2(16)
COMMON /TIMDAT/ TI(1500),IAREA(1500),TEND,NTIME,NEWTIM,TDEL
DOUBLE PRECISION TI
COMMON /SYSAVG/ QSAVG,QFAVG,QTAVG,QRAVG,ILOPT
COMMON /QPARAM/ NAME(100),LAMDA(100),OLDLAM(100),TEST2(100),
1 TEST1(100),TAU(100),REPAIR(100),QOVRD(100),
2 PTCF(100),INEFF(100),ULAMDA(100),QRESID(100),
3 BETA(100),CARDVR(100),OFFSET(100),FNLAMD(100),
4 FNQRSD(100),FTQRSD(100),ITYPE(100),NCOMP,DIFF
REAL LAMDA,INEFF
DOUBLE PRECISION NAME
COMMON /QCON/ NTEST(100),PTCF1(100),QRES1(100),QN(100),
1 QUN(100),PK(100,100)
C
C THIS SECTION IS CHANGED BY THE USER TO SUIT HIS REQUIREMENTS
TST=TEST2(1)/24.
OFSET=OFFSET(1)/(24.*365.)
IF(NRUN.GT.1) GO TO 9
WRITE(10,11)
11 FORMAT(' LAMBDA(1) OFFSET(1) BETA(1) QRESID(1) TEST2(1) QSAVG')
9 WRITE(10,2) LAMDA(1),OFSET,BETA(1),QRESID(1),TST,QSAVG
2 FORMAT(6(1PE10.2))
RETURN
END
```

It is frequently convenient to keep more than one FILES subroutine. This can be done by creating them under unique filenames. When any specific subroutine is required, a series of commands will change the active FILES subroutine to the one desired. For example, the commands:

```
x initfile fortran
file files
R: T=0.07/0.18 00:43:19
fortgi files
G1 COMPILER ENTERED
SOURCE ANALYZED
PROGRAM NAME = FILES
* NO DIAGNOSTICS GENERATED
R: T=0.27/0.48 00:43:30
```

call the filename INITFILE FORTRAN and files a copy as FILES FORTRAN. After it is compiled the formatted output in the FILE will follow these Fortran statements:

```
TST5=TEST2(5)/24.  
TST86=TEST2(86)/24.  
IF(NRUN.GT.1) GO TO 9  
WRITE(10,11)  
11 FORMAT(' LAMBDA(6) LAMBDA(12)TAU(5) TEST2(5) TEST2(86) QSAVG')  
9 RT=OFFSET(1)/(24.*365.)  
WRITE(10,2) LAMDA(6),LAMDA(12),TAU(5),TST5,TST86,QSAVG  
2 FORMAT(6(1PE10.2))  
RETURN  
END
```

NUCLEAR ENGINEERING
READING ROOM - M.I.T.