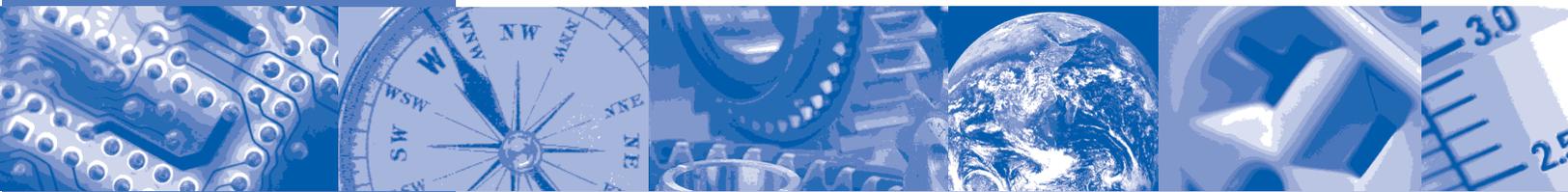


ISA-TR84.00.02-2002 - Part 5



**Safety Instrumented Functions
(SIF)-Safety Integrity Level (SIL)
Evaluation Techniques Part 5:
Determining the PFD of SIS Logic
Solvers via Markov Analysis**



**ISA—The Instrumentation,
Systems, and
Automation Society**

Approved 17 June 2002

ISA-TR84.00.02-2002 – Part 5

Safety Instrumented Functions (SIF) — Safety Integrity Level (SIL) Evaluation Techniques Part 5:
Determining the PFD of SIS Logic Solvers via Markov Analysis

ISBN: 1-55617-806-9

Copyright © 2002 by The Instrumentation, Systems, and Automation Society. All rights reserved. Not for resale. Printed in the United States of America. No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means (electronic mechanical, photocopying, recording, or otherwise), without the prior written permission of the Publisher.

ISA
67 Alexander Drive
P.O. Box 12277
Research Triangle Park, North Carolina 27709

Preface

This preface, as well as all footnotes and annexes, is included for information purposes and is not part of ISA-TR84.00.02-2002 – Part 5.

This document has been prepared as part of the service of ISA—the Instrumentation, Systems, and Automation Society—toward a goal of uniformity in the field of instrumentation. To be of real value, this document should not be static but should be subject to periodic review. Toward this end, the Society welcomes all comments and criticisms and asks that they be addressed to the Secretary, Standards and Practices Board; ISA; 67 Alexander Drive; P. O. Box 12277; Research Triangle Park, NC 27709; Telephone (919) 549-8411; Fax (919) 549-8288; E-mail: standards@isa.org.

The ISA Standards and Practices Department is aware of the growing need for attention to the metric system of units in general, and the International System of Units (SI) in particular, in the preparation of instrumentation standards. The Department is further aware of the benefits to USA users of ISA standards of incorporating suitable references to the SI (and the metric system) in their business and professional dealings with other countries. Toward this end, this Department will endeavor to introduce SI-acceptable metric units in all new and revised standards, recommended practices, and technical reports to the greatest extent possible. *Standard for Use of the International System of Units (SI): The Modern Metric System*, published by the American Society for Testing & Materials as IEEE/ASTM SI 10-97, and future revisions, will be the reference guide for definitions, symbols, abbreviations, and conversion factors.

It is the policy of ISA to encourage and welcome the participation of all concerned individuals and interests in the development of ISA standards, recommended practices, and technical reports. Participation in the ISA standards-making process by an individual in no way constitutes endorsement by the employer of that individual, of ISA, or of any of the standards, recommended practices, and technical reports that ISA develops.

CAUTION — ISA ADHERES TO THE POLICY OF THE AMERICAN NATIONAL STANDARDS INSTITUTE WITH REGARD TO PATENTS. IF ISA IS INFORMED OF AN EXISTING PATENT THAT IS REQUIRED FOR USE OF THE STANDARD, IT WILL REQUIRE THE OWNER OF THE PATENT TO EITHER GRANT A ROYALTY-FREE LICENSE FOR USE OF THE PATENT BY USERS COMPLYING WITH THE STANDARD OR A LICENSE ON REASONABLE TERMS AND CONDITIONS THAT ARE FREE FROM UNFAIR DISCRIMINATION.

EVEN IF ISA IS UNAWARE OF ANY PATENT COVERING THIS STANDARD, THE USER IS CAUTIONED THAT IMPLEMENTATION OF THE STANDARD MAY REQUIRE USE OF TECHNIQUES, PROCESSES, OR MATERIALS COVERED BY PATENT RIGHTS. ISA TAKES NO POSITION ON THE EXISTENCE OR VALIDITY OF ANY PATENT RIGHTS THAT MAY BE INVOLVED IN IMPLEMENTING THE STANDARD. ISA IS NOT RESPONSIBLE FOR IDENTIFYING ALL PATENTS THAT MAY REQUIRE A LICENSE BEFORE IMPLEMENTATION OF THE STANDARD OR FOR INVESTIGATING THE VALIDITY OR SCOPE OF ANY PATENTS BROUGHT TO ITS ATTENTION. THE USER SHOULD CAREFULLY INVESTIGATE RELEVANT PATENTS BEFORE USING THE STANDARD FOR THE USER'S INTENDED APPLICATION.

HOWEVER, ISA ASKS THAT ANYONE REVIEWING THIS STANDARD WHO IS AWARE OF ANY PATENTS THAT MAY IMPACT IMPLEMENTATION OF THE STANDARD NOTIFY THE ISA STANDARDS AND PRACTICES DEPARTMENT OF THE PATENT AND ITS OWNER.

ADDITIONALLY, THE USE OF THIS STANDARD MAY INVOLVE HAZARDOUS MATERIALS, OPERATIONS OR EQUIPMENT. THE STANDARD CANNOT ANTICIPATE ALL POSSIBLE APPLICATIONS OR ADDRESS ALL POSSIBLE SAFETY ISSUES ASSOCIATED WITH USE IN HAZARDOUS CONDITIONS. THE USER OF THIS STANDARD MUST EXERCISE SOUND

PROFESSIONAL JUDGMENT CONCERNING ITS USE AND APPLICABILITY UNDER THE USER'S PARTICULAR CIRCUMSTANCES. THE USER MUST ALSO CONSIDER THE APPLICABILITY OF ANY GOVERNMENTAL REGULATORY LIMITATIONS AND ESTABLISHED SAFETY AND HEALTH PRACTICES BEFORE IMPLEMENTING THIS STANDARD.

THE USER OF THIS DOCUMENT SHOULD BE AWARE THAT THIS DOCUMENT MAY BE IMPACTED BY ELECTRONIC SECURITY ISSUES. THE COMMITTEE HAS NOT YET ADDRESSED THE POTENTIAL ISSUES IN THIS VERSION.

The following people served as members of ISA Committee SP84:

NAME	COMPANY
V. Maggioli, Chair	Feltronics Corporation
R. Webb, Managing Director	POWER Engineers
C. Ackerman	Air Products & Chemicals Inc.
R. Adamski	Invensys
C. Adler	Moore Industries International Inc.
R. Bailliet	Syscon International Inc.
N. Battikha	Bergo Tech Inc.
L. Beckman	HIMA Americas Inc.
S. Bender	S K Bender & Associates
K. Bond	Shell Global Solutions
A. Brombacher	Eindhoven University of Technology
S. Brown*	DuPont Company
J. Carew	Consultant
K. Dejmek	Baker Engineering & Lisk Consulting
A. Dowell*	Rohm & Haas Company
R. Dunn*	DuPont Engineering
P. Early	ABB Industrial Systems Inc.
T. Fisher	Deceased
J. Flynt	Consultant
A. Frederickson	Triconex Corporation
R. Freeman	ABS Consulting
D. Fritsch	Fritsch Consulting Service
K. Gandhi	Kellogg Brown & Root
R. Gardner*	Dupont
J. Gilman	Consultant
W. Goble	exida.com LLC
D. Green*	Rohm & Haas Company
P. Gruhn	Siemens
C. Hardin	CDH Consulting Inc.
J. Harris	UOP LLC
D. Haysley	Albert Garaody & Associates
M. Houtermans	TUV Product Service Inc.
J. Jamison	Bantrel Inc.
W. Johnson*	E I du Pont
D. Karydas*	Factory Mutual Research Corporation
L. Laskowski	Solutia Inc.
T. Layer	Emerson Process Management
D. Leonard	D J Leonard Consultants
E. Lewis	Consultant
E. Marszal	Exida.com
N. McLeod	Atofina
W. Mostia	WLM Engineering Company
D. Ogwude	Creative Systems International

G. Ramachandran	Cytec Industries Inc.
K. Schilowsky	Marathon Ashland Petroleum Company LLC
D. Sniezek	Lockheed Martin Federal Services
C. Sossman	WG-W Safety Management Solutions
R. Spiker	Yokogawa Industrial Safety Systems BV
P. Stavrianidis*	Factory Mutual Research Corporation
H. Storey	Equilon Enterprises LLC
A. Summers	SIS-TECH Solutions LLC
L. Suttinger	Westinghouse Savannah River Company
R. Szanyi	ExxonMobil Research Engineering
R. Taubert	BASF Corporation
H. Tausch	Honeywell Inc.
T. Walczak	GE FANUC Automation
M. Weber	System Safety Inc.
D. Zetterberg	Chevron Texaco ERTC

* One vote per company.

This standard was approved for publication by the ISA Standards and Practices Board on 17 June 2002.

NAME	COMPANY
M. Zielinski	Emerson Process Management
D. Bishop	David N Bishop, Consultant
D. Bouchard	Paprican
M. Cohen	Consultant
M. Coppler	Ametek, Inc.
B. Dumortier	Schneider Electric
W. Holland	Southern Company
E. Icyan	ACES Inc
A. Iverson	Ivy Optiks
R. Jones	Dow Chemical Company
V. Maggioli	Feltronics Corporation
T. McAviney	ForeRunner Corporation
A. McCauley, Jr.	Chagrin Valley Controls, Inc.
G. McFarland	Westinghouse Process Control Inc.
R. Reimer	Rockwell Automation
J. Rennie	Factory Mutual Research Corporation
H. Sasajima	Yamatake Corporation
I. Verhappen	Syncrude Canada Ltd.
R. Webb	POWER Engineers
W. Weidman	Parsons Energy & Chemicals Group
J. Weiss	KEMA Consulting
M. Widmeyer	Stanford Linear Accelerator Center
C. Williams	Eastman Kodak Company
G. Wood	Graeme Wood Consulting

This page intentionally left blank.

Contents

Foreword.....	9
Introduction	11
1 Scope.....	17
2 References	17
3 Definitions	18
4 Logic solver modeling using Markov analysis	18
4.1 Probability of Failure on Demand (PFD)	18
4.2 Markov modeling methodology	19
4.3 Assumptions and limitations.....	21
4.4 Basic Markov model description.....	22
5 Procedures for quantification of logic solver performance	23
5.1 Assumptions and limitations.....	23
5.2 Calculations and reports.....	25
6 Logic solver Markov models calculation results	25
6.1 Description and results of the reliability calculation for three E/E/PE logic solver configurations including input data tables.....	26
6.2 Configuration drawings, Markov diagrams and calculation results	27
Annex A (informative) — Markov model development and quantification.....	49
Annex B (informative) — Logic solver model input data.....	91
Annex C — Index.....	99

This page intentionally left blank.

Safety Instrumented Functions (SIF)

— Safety Integrity Level (SIL) Evaluation Techniques

Part 5: Determining the PFD of Logic Solvers via Markov Analysis

Foreword

The information contained in ISA-TR84.00.02-2002 – Part 5 is provided for information only and is not part of the ANSI/ISA-84.01-1996 Standard⁽¹⁾ requirements.

The purpose of ISA-TR84.00.02-2002⁽²⁾ is to provide the process industry with a description of various methodologies that can be used to evaluate the Safety Integrity Level (SIL) of Safety Instrumented Systems (SIS).

ANSI/ISA-84.01-1996 provides the minimum requirements for implementing a SIS given that a set of functional requirements have been defined and a SIL requirement has been established for each safety function. Additional information of an informative nature is provided in the annexes to ANSI/ISA-84.01-1996 to assist the designer in applying the concepts necessary to achieve an acceptable design. However, Standards Project 84 (SP84) determined that it was appropriate to provide supplemental information that would assist the user in evaluating the capability of any given SIS design to achieve its required SIL. A secondary purpose of this document is to reinforce the concept of the performance based evaluation of SIS. The performance parameters that satisfactorily service the process industry are derived from the SIL and reliability evaluation of SIS, namely the probability of the SIS to fail to respond to a demand and the probability that the SIS creates a nuisance trip. Such evaluation addresses the design elements (hardware, software, redundancy, etc.) and the operational attributes (inspection/maintenance policy, frequency and quality of testing, etc.) of the SIS. The basis for the performance evaluation of the SIS is safety targets determined through hazard analysis and risk assessment⁽⁶⁾ of the process. This document demonstrates methodologies for determining the SIL and the probability of spurious trip of the SIS.

The document focuses on methodologies that can be used without promoting a single methodology. It provides information on the benefits of various methodologies as well as some of the drawbacks they may have.

THE METHODOLOGIES ARE DEMONSTRATED THROUGH EXAMPLES (SIS ARCHITECTURES) THAT REPRESENT POSSIBLE SYSTEM CONFIGURATIONS AND SHOULD NOT BE INTERPRETED AS RECOMMENDATIONS FOR SIS. THE USER IS CAUTIONED TO CLEARLY UNDERSTAND THE ASSUMPTIONS AND DATA ASSOCIATED WITH THE METHODOLOGIES IN THIS DOCUMENT BEFORE ATTEMPTING TO UTILIZE THE METHODS PRESENTED HEREIN.

The users of ISA-TR84.00.02-2002 include:

- Process Hazards Analysis teams that wish to develop understanding of different methodologies in determining SIL
- SIS designers who want a better understanding of how redundancy, diagnostic coverage, diversity, etc., fit into the development of a proper SIS architecture
- Logic solver and field device suppliers

- National and International standard bodies providing guidance in the use of reliability techniques for SIS architectures
- Reliability engineers (or any engineer performing this function) can use this information to develop better methods for determining SIL in the rapidly changing SIS field
- Parties who do not have a large installed base of operating equipment sufficient to establish appropriate statistical analysis for PFD_{avg} and $MTTF^{spurious}$ for SIS components
- Operations and maintenance personnel

ISA-TR84.00.02-2002 consists of the following parts, under the general title "Safety Instrumented Systems (SIS) — Safety Integrity Level (SIL) Evaluation Techniques."

Part 1: Introduction

Part 2: Determining the SIL of a SIF via Simplified Equations

Part 3: Determining the SIL of a SIF via Fault Tree Analysis

Part 4: Determining the SIL of a SIF via Markov Analysis

Part 5: Determining the PFD of SIS Logic Solvers via Markov Analysis

Introduction

ANSI/ISA-84.01-1996 describes a safety life cycle model for the implementation of risk reduction measures for the process industry (Clause 4). The standard then proceeds to provide specific guidance in the application of SIS, which may be one of the risk reduction methods used. The standard defines three levels of safety integrity (Safety Integrity Levels, SIL) that may be used to specify the capability that a safety function must achieve to accomplish the required risk reduction. ISA-TR84.00.02-2002 provides methodologies for evaluating SIS to determine if they achieve the specific SIL. This may be referred to as a probability of failure on demand (PFD) evaluation of the SIS.

ISA-TR84.00.02-2002 only addresses SIS operating in demand mode.

The evaluation approaches outlined in this document are performance-based approaches and do not provide specific results that can be used to select a specific architectural configuration for a given SIL.

THE READER IS CAUTIONED TO CLEARLY UNDERSTAND THE ASSUMPTIONS ASSOCIATED WITH THE METHODOLOGY AND EXAMPLES IN THIS DOCUMENT BEFORE DERIVING ANY CONCLUSIONS REGARDING THE EVALUATION OF ANY SPECIFIC SIS.

The evaluation processes described in this document take place before the SIS detailed design phase of the life cycle (see Figure I.1, Safety Life Cycle Model).

This document assumes that a SIS is required. It does not provide guidance in the determination of the need for a SIS. The user is referred to ANSI/ISA-84.01-1996 Annex A for methodologies that might be used in making this determination.

This document involves the evaluation of the whole SIS from the sensors through the logic solver to the final elements. Process industry experience shows that sensors and final elements are major contributors to loss of SIS integrity (high PFD). When evaluating the performance of sensors and final elements, issues such as component technology, installation, and maintenance should be considered.

Frequently, multiple safety functions are included in a single logic solver. Generally, the safety function case with the highest SIL requirement will be the case that determines whether the logic solver meets performance requirements. When multiple safety function cases have the same SIL, select the case with the largest number of I/O, number of I/O channels, etc., to determine whether the logic solver meets performance requirements. The logic solver should be carefully evaluated since a problem in the logic solver may adversely impact the performance of all of the safety functions (e.g., common cause).

This principle (e.g., common cause) applies to any

- element of a SIS that is common to more than one safety function; and
- redundant element with one or more safety function.

Each element should be evaluated with respect to all the safety functions with which it is associated

- to ensure that it meets the integrity level required for each safety function;
- to understand the interactions of all the safety functions; and
- to understand the impact of failure of each component.

This document does not provide guidance in the determination of the specific SIL required (e.g., SIL 1, 2, 3) for the SIS. The user is again referred to ANSI/ISA-84.01-1996 or to other references.

The primary focus of this document is on evaluation methodologies for assessing the capability of the SIS. To understand what is meant by the SIS, refer to the model defined in ANSI/ISA-84.01-1996 and repeated in Figure I.2 defining the boundaries of the SIS.

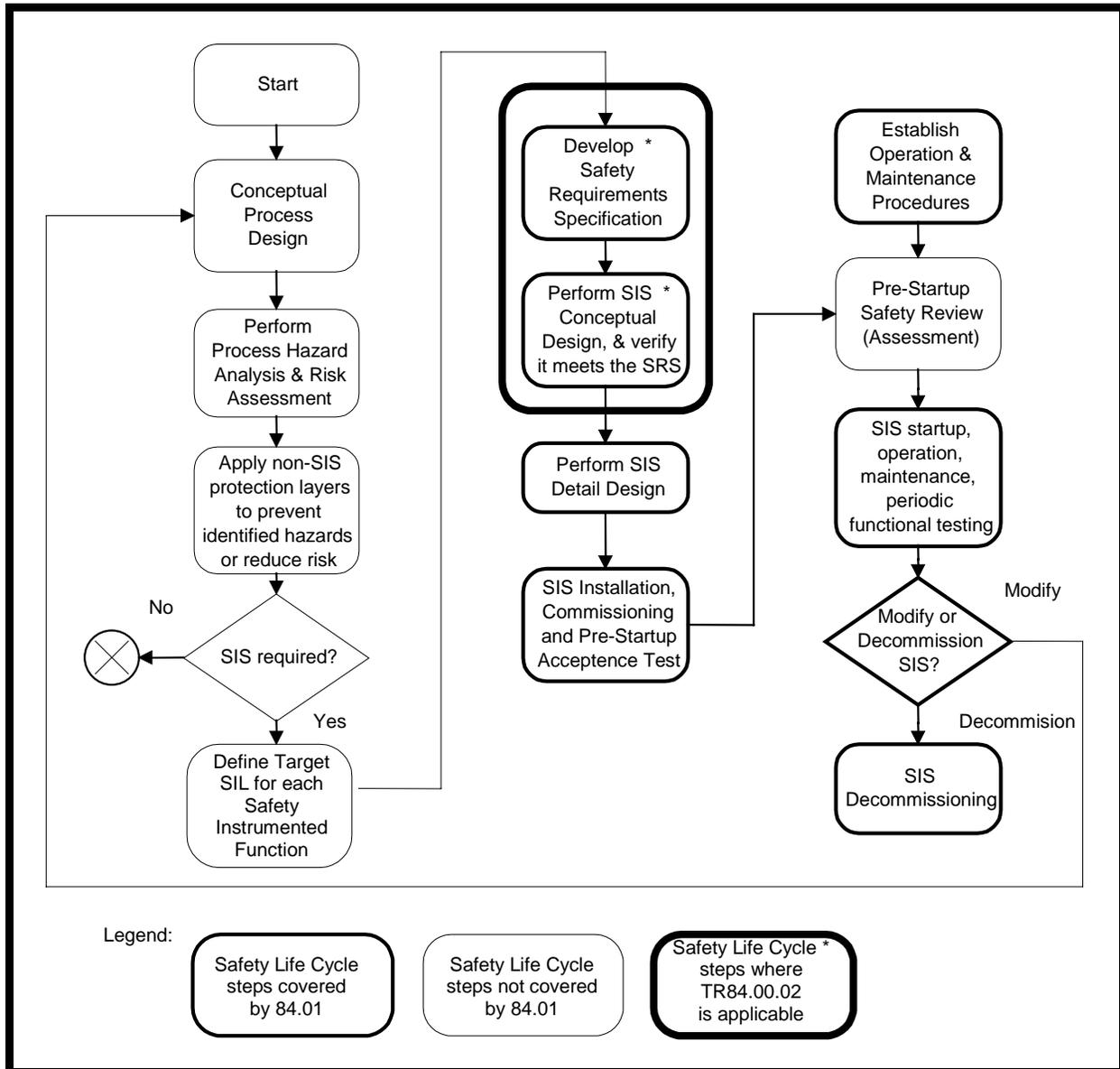


Figure I.1 — Safety life cycle model

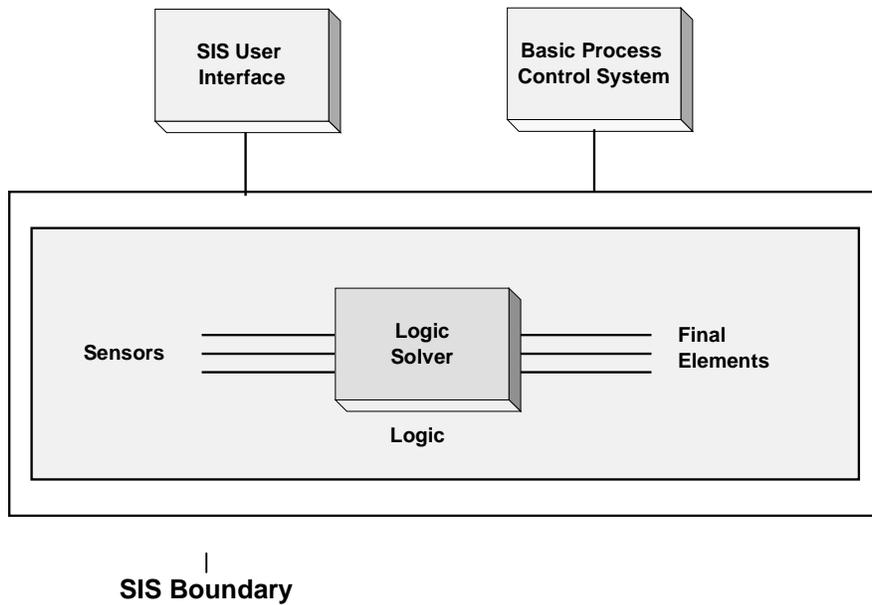


Figure I.2 — Definition of Safety Instrumented Systems (SIS)

The SIS Safety Requirements address the design elements (hardware, software, redundancy, etc.) and the operational attributes (inspection/maintenance policy, frequency and quality of testing, etc.) of the SIS. These elements are used to evaluate the PFD of each safety function.

The PFD of these systems can be determined using historical system performance data (e.g., statistical analysis). Where systems, subsystems, components, etc. have not been in use for a sufficiently long time and in large enough numbers to have a statistically significant population available for the evaluation of their performance solely based on actuarial data, a systematic evaluation of the performance of a system may be obtained through the use of PFD analysis techniques.

PFD analysis techniques employ systematic methodologies that decompose a complex system to its basic components. The performance and interactions of these basic components are merged into reliability models (such as simplified equations, fault trees, Markov models) to determine the overall system safety availability.

This document provides users with a number of PFD evaluation techniques, which allow a user to determine if a SIS meets the required safety integrity levels.

Safety integrity is defined as “The probability of a Safety Instrumented System satisfactorily performing the required safety functions under all stated conditions within a stated period of time.” Safety integrity consists of two elements: 1) hardware safety integrity and 2) systematic safety integrity. Hardware safety integrity, which is based upon random hardware failures, can normally be estimated to a reasonable level of accuracy. ANSI/ISA-84.01-1996 addresses the hardware and systematic safety integrity by specifying target failure measures for each SIL. For SIS operating in the demand mode the target failure measure is **PFD_{avg}** (average probability of failure to perform its design function on demand). **PFD_{avg}** is also commonly referred to as the average probability of failure on demand. Systematic integrity is difficult to quantify due to the diversity of causes of failures; systematic failures may be introduced during the specification, design, implementation, operational and modification phases and may affect hardware as

well as software. ANSI/ISA-84.01-1996 addresses systematic safety integrity by specifying procedures, techniques, measures, etc. that reduce systematic failures.

An acceptable safe failure rate is also normally specified for a SIS. The safe failure rate is commonly referred to as the false trip, nuisance trip, or spurious trip rate. The spurious trip rate is included in the evaluation of a SIS, since process start up and shutdown are frequently periods where chances of a hazardous event are high. Hence in many cases, the reduction of spurious trips will increase the safety of the process. The acceptable safe failure rate is typically expressed as the mean time to a spurious trip (**MTTF^{spurious}**).

NOTE In addition to the safety issue(s) associated with spurious trips the user of the SIS may also want the acceptable **MTTF^{spurious}** to be increased to reduce the effect of spurious trips on the productivity of the process under control. This increase in the acceptable **MTTF^{spurious}** can usually be justified because of the high cost associated with a spurious trip.

The objective of this technical report is to provide users with techniques for the evaluation of the hardware and systematic safety integrity of SIS (**PFD_{avg}**) and the determination of **MTTF^{spurious}**. The three methods in this technical report allow modeling of both systematic failures so that a quantitative analysis can be performed.

ISA-TR84.00.02-2002 shows how to model complete SIF, which include the sensors, the logic solver and final elements. To the extent possible the system analysis techniques allow these elements to be independently analyzed. This allows the SIS designer to select the proper system configuration to achieve the required safety integrity level.

ISA-TR84.00.02-2002 - Part 1 provides

- a detailed listing of the definition of all terms used in this document. These are consistent with the ANSI/ISA-84.01-1996, IEC 61508 and IEC 61511 standards.
- the background information on how to model all the elements or components of a SIF. It focuses on the hardware components, provides some component failure rate data that are used in the examples calculations and discusses other important parameters such as common cause failures and functional failures.
- a brief introduction to the methodologies that will be used in the examples shown in this document. They are Simplified equations ⁽³⁾, Fault Tree Analysis ⁽⁴⁾, and Markov Analysis ⁽⁵⁾.

ISA-TR84.00.02-2002 - Part 2 provides simplified equations for calculating the SIL values for Demand Mode Safety Instrumented Functions (SIF) installed in accordance with ANSI/ISA-84.01-1996, "Applications of Safety Instrumented Systems for the Process Industries." Part 2 should not be interpreted as the only evaluation technique that might be used. It does, however, provide the engineer(s) performing design for a SIS with an overall technique for assessing the capability of the designed SIF.

ISA-TR84.00.02-2002 - Part 3 provides fault tree analysis techniques for calculating the SIL for Demand Mode Safety Instrumented Functions (SIF) installed in accordance with ANSI/ISA-84.01-1996, "Applications of Safety Instrumented Systems for the Process Industries." Part 3 should not be interpreted as the only evaluation technique that might be used. It does, however, provide the engineer(s) performing design for a SIS with an overall technique for assessing the capability of the designed SIF.

ISA-TR84.00.02-2002 - Part 4 provides Markov analysis techniques for calculating the SIL values for Demand Mode Safety Instrumented Functions (SIF) installed in accordance with ANSI/ISA-84.01-1996, "Applications of Safety Instrumented Systems for the Process Industries." Part 4 should not be interpreted as the only evaluation technique that might be used. It does, however, provide the

engineer(s) performing design for a SIS with an overall technique for assessing the capability of the designed SIF.

ISA-TR84.00.02-2002 - Part 5 addresses the logic solver only, using Markov Models for calculating the PFD of E/E/PE logic solvers because it allows the modeling of maintenance and repairs as a function of time, treats time as a model parameter, explicitly allows the treatment of diagnostic coverage, and models the systematic failures (i.e., operator failures, software failures, etc.) and common cause failures.

Figure I.3 illustrates the relationship of each part to all other parts.

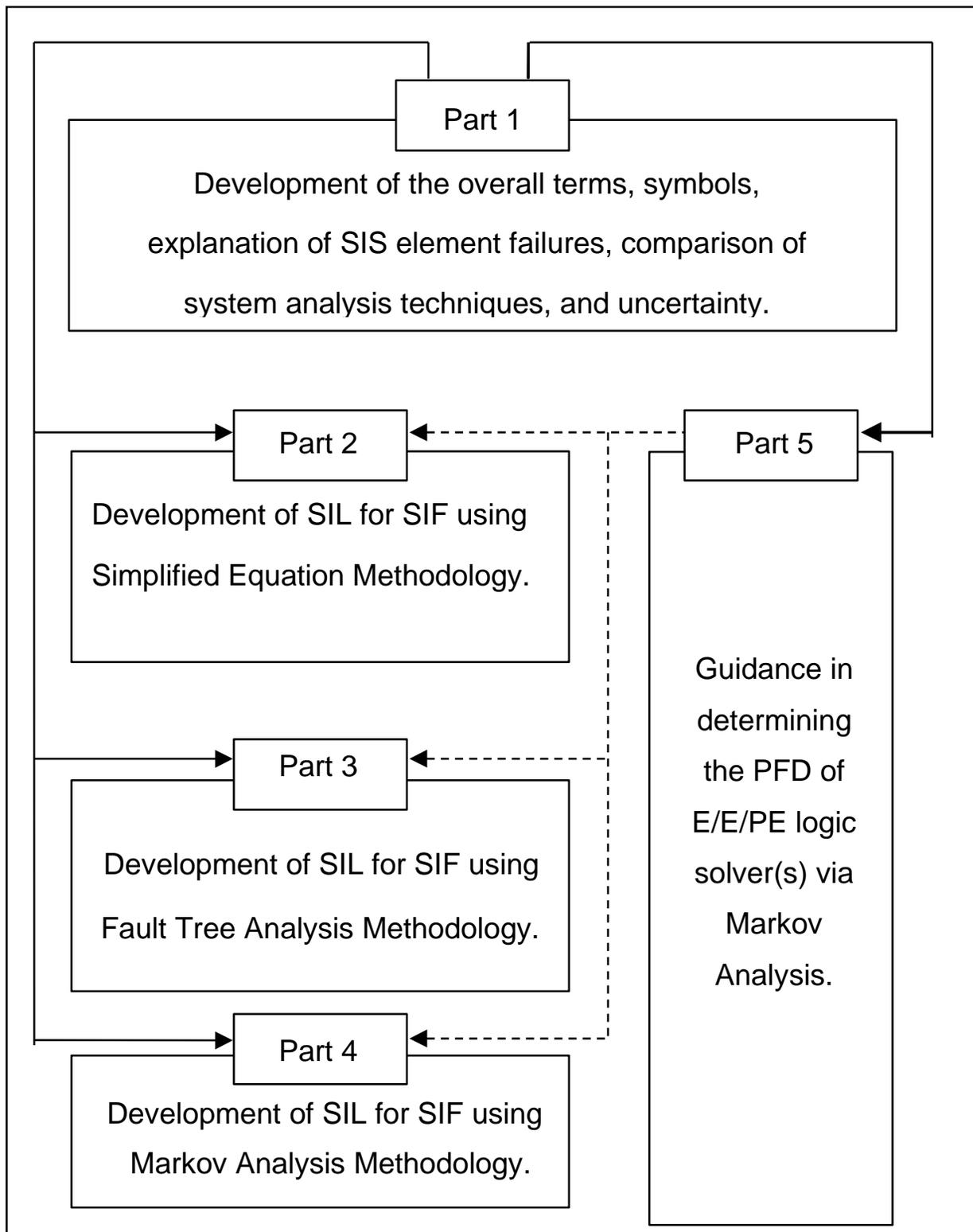


Figure I.3 — ISA-TR84.00.02-2002 overall framework

1 Scope

1.1 ISA-TR84.00.02-2002 - Part 5 is informative and does not contain any mandatory requirements. ISA-TR84.00.02-2002 - Part 5 is intended to be used only with a thorough understanding of ISA-TR84.00.02-2002 - Part 1 which defines the overall scope.

1.2 ISA-TR84.00.02-2002 - Part 5 provides:

a) guidance in PFD analysis of logic solvers;

NOTE The term "logic solver" will be used throughout Part 5 to indicate the SIS logic solver. The logic solver technology may be any E/E/PES.

b) a method to determine the PFD of logic solvers;

c) failure rates and failure modes of logic solvers;

d) the impact of diagnostics, diagnostic coverage, covert faults, test intervals, common cause, systematic failures, redundancy of logic solvers on the PFD of the logic solver; and

e) a method for the verification of PFD of logic solvers.

1.3 The procedures and examples outlined in ISA-TR84.00.02-2002 - Part 5 provide the engineer with Markov modeling steps to be followed in determining a mathematical value for the PFD for typical configurations of SIS logic solvers designed according to ANSI/ISA-84.01-1996.

1.4 Persons using ISA-TR84.00.02-2002 - Part 5 require a basic knowledge of Markov Analysis.

1.5 See ISA-TR84.00.02-2002 - Part 1 (Introduction), Part 2 (Simplified Equations), Part 3 (Fault Tree Analysis), and Part 4 (Markov Analysis) if it is necessary to mathematically evaluate the SIL of the safety instrumented function (SIF).

NOTE The method illustrated herein (i. e. Markov analysis) may also be used to determine the PFD of other SIF components such as sensors and final elements. The logic solver was selected to illustrate how Markov Analysis is applied to a complex SIF component.

2 References

1. ANSI/ISA-84.01-1996 "Application of Safety Instrumented Systems for the Process Industries," Instrumentation, Systems, and Automation Society, Research Triangle Park, NC, 27709, February 1996.
2. ISA-TR84.00.02-2002, "Safety Instrumented Functions (SIF) – Safety Integrity Level Evaluation Techniques, Part 1: Introduction; Part 2: Determining the SIL of a SIF via Simplified Equations; Part 3: Determining the SIL of a SIF via Fault Tree Analysis; Part 4: Determining the SIL of a SIF via Markov Analysis; Part 5: Determining the PFD of SIS Logic Solvers via Markov Analysis," Instrumentation, Systems and Automation Society, Technical Report, Research Triangle Park, NC, 27709, 2002.
3. "Reliability, Maintainability and Risk" by David J. Smith, 4th Edition, 1993, Butterworth-Heinemann, ISBN 82-515-0188-1.
4. "Guidelines for Safe Automation of Chemical Processes," Center for Chemical Process Safety, American Institute of Chemical Engineers, New York, NY 10017, 1993.

5. "Evaluating Control Systems Reliability," W. M. Goble, Instrument Society of America, Research Triangle Park, NC, 27709, 1990.
6. "Probabilistic Risk Assessment," Henley, Ernest J. and Kumamoto, Kiromitsu, IEEE Press, New York, New York, 1992.
7. "Reliability by Design," A.C. Brombacher, John Wiley & Sons, New York, NY 10158, 1992.
8. "Software Reliability Handbook," P. Rook, Elsevier Science Press, New York, NY 10010, 1990.
9. "Introduction to Reliability Engineering," E.E. Lewis, John Wiley & Sons, New York, NY 10158, 1987.
10. "Reliability Evaluation of Engineering Systems," R. Billinton, R.N. Allan, Pitman Advanced Publishing Program, Marshfield, MA 02050, 1983.

3 Definitions

Definitions and terminology used in this part are defined in ISA-TR84.00.02-2002 – Part 1.

4 Logic solver modeling using Markov analysis

ISA-TR84.00.02-2002 – Part 5 focuses on the logic solver associated with the safety instrumented function (SIF).

The objective is to develop the reliability models for three logic solver architectures using Markov analysis. The quantification of the models will produce the desired logic solver performance parameters:

- a) The probability to fail on demand (PFD), and
- b) The probability to fail spurious (PFS).

The Markov technique has been explained in ISA-TR84.00.02-2002 - Part 4 (Determining the SIL of a SIF using Markov Analysis). The reader who is interested in learning more about Markov modeling is referred to

- a) Evaluating Control Systems Reliability⁽⁵⁾, Chapter 5;
- b) Reliability Evaluation of Engineering Systems⁽¹⁰⁾, Chapters 8 and 9;
- c) Introduction to Reliability Engineering⁽⁹⁾, Chapter 9; and
- d) ISA-TR84.00.02-2002 – Part 4.

4.1 Probability of Failure on Demand (PFD)

See ISA-TR84.00.02-2002 - Part 1, Clause 4 for information on the probability of failure on demand (PFD).

4.2 Markov modeling methodology

Markov models^(7,9,10) are created by identifying all the possible states that the logic solver may enter while transitioning from fully operational, through partially failed (degraded) states, to the failed state. To accomplish this task, the different logic solver states are identified during the failure modes and effects analysis (FMEA) and the corresponding transition probabilities (i.e., probabilities of components that must fail in order to transition from one state to another state) are shown as arcs on the Markov model.

Markov model construction starts with the state of the logic solver where all of the components are functioning properly (successful state). To develop the other states, the following general procedure is followed:

For any state

- a) list all of the logic solver components, and
- b) list the ways the logic solver components may leave that state. There are two ways that a component can leave a state.
 - 1) First, a component in an operating state can fail.
 - 2) Second, a component in a failed state can be repaired.

In the former case, the probability of a component failure is the driving mechanism to force a transition out of the state. For exponential failure and repair probability distributions and using the rare event approximation (ISA-TR84.00.02-2002 – Part 5, Annex A, Clause A.4, Equation 4) the probability of failure is defined as λt , where λ is the failure rate of the component and t is the time. For the latter case, the repair probability is given as μt , where μ is the repair rate. Due to convention, these probabilities in the Markov models are shown as simply failure rates and/or repair rates and are commonly referred to as transition rates. The transition probabilities are always considered in the formulation and analysis of the models.

Annex A illustrates how a Markov model is created for the logic solver shown in Figure 4.1 which is a Dual PE logic solver having Dual Input and Dual Output modules, with One-out-of-Two (1oo2) shutdown logic.

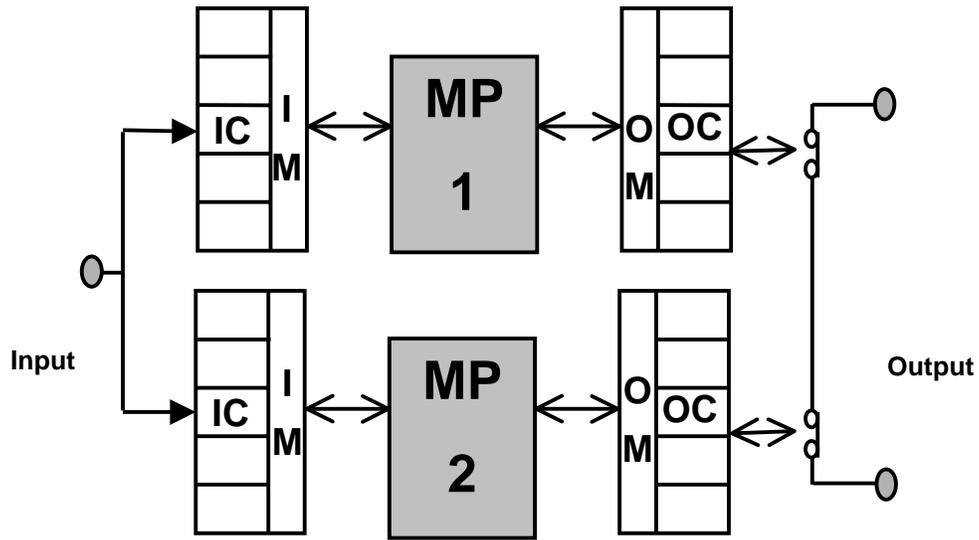


Figure 4.1 — Hypothetical - dual PE logic solver with dual I/O, one-out-of-two (1oo2) shutdown logic

Before a Markov Model can be developed an FMEA is typically performed to determine the hardware failure rates. The failure rate must be broken down into safe and dangerous fractions, so the complete performance of the logic solver can be evaluated. In fact, the safe and dangerous failure rates should also be broken down into the detected and undetected failures, as determined by the logic solver on-line diagnostics. Hence the FMEA should result in the determination of the component failure categories shown in Table 4.1. The FMEA method is described in ISA-TR84.00.02-2002 - Part 1, Annex D.

Table 4.1 — Component failure categories

Component	Failure Categories					
IP	SCC	DCC	SD	SU	DD	DU
OP	SCC	DCC	SD	SU	DD	DU
MP	SCC	DCC	SD	SU	DD	DU
IC	SCC	DCC	SD	SU	DD	DU
OC	SCC	DCC	SD	SU	DD	DU

<p>Legend: Component</p> <p>IP – Common part of input module</p> <p>OP – Common part of output module</p> <p>MP – Main Processor</p> <p>IC – Input Channel</p> <p>OC – Output Channel</p> <p>Failure Categories</p> <p>SCC – Safe common cause failure</p> <p>DCC – Dangerous common cause failure</p> <p>SD – Safe detected hardware failure</p> <p>SU – Safe undetected hardware failure</p> <p>DD – Dangerous detected hardware failure</p> <p>DU – Dangerous undetected hardware failure</p>

4.3 Assumptions and limitations

All of the assumptions made while developing the models are listed below. The impact on the models if these assumptions are changed is also discussed.

1. The calculations are based on de-energize-to-trip SIFs.
2. Failure rates and on-line repair rates are assumed to be constant.
3. The mission time for the logic solver is assumed to be the time between function tests of the logic solver. This assumption eliminates the repair of any undetected failures because they would only be detected during the functional test interval.
4. The hazard and risk analysis shall define the acceptable response in the event of loss of power.
5. One type of input module and one type of output module are used in all Markov models. It is assumed that there are **n** input modules and **m** output modules in each leg of every architecture. Additional module types can be easily included by modifying the models to account for the additional safe and dangerous failure transitions.
6. It is assumed that plant personnel will initiate action to take the process to a safe state when a dangerous failure is detected in the logic solver (operator response is assumed to be before a demand occurs, i.e., instantaneous, and PFD of operator response is assumed to be 0).

NOTE If the action depends on the plant personnel to provide safety, the user is cautioned to account for the probability of failure of personnel to perform the required function in a timely manner.

7. The models assume that the inspection and repair functions that are performed are perfect and bring the logic solver to a "as good as new state."
8. Channels in multi-channel architectures are treated as completely independent.
9. The data used for the example calculations can be found in Annex B.

4.4 Basic Markov model description

The basic Markov model used for the logic solver architectures is given in Figure 4.2.

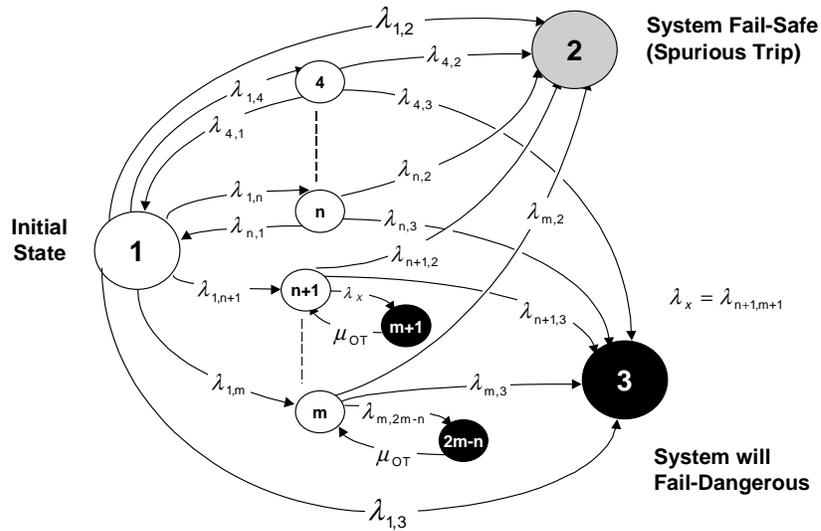


Figure 4.2 — Basic Markov m model

The basic Markov model has the following final and intermediate (transition) states:

- 1: The initial state. No failures are present in the logic solver.
- 2: The fail-safe state. The logic solver is in the shutdown safe state.
- 3: The fail-dangerous state. The logic solver will not respond to a process demand.
- 4...n: The intermediate states where one or more (safe detected, safe undetected or dangerous detected) failures are present without being in the fail-safe state.
- n+1...m: The intermediate safe states where one or more dangerous undetected failures are present without being in the fail-dangerous state.
- m+1...2m-n: The intermediate fail-dangerous states where a combination of dangerous undetected and a safe or dangerous detected fault is present that can be repaired bringing the logic solver back to state n+1+m. If a demand comes in this situation the logic solver will not be able to perform the safety function.

The transitions in this model are:

- $\lambda_{1,2}$ Immediate transition from the initial-state to the fail-safe state. Caused by safe failures in single components (e.g., single inputs) and by safe common cause failures.

$\lambda_{1,3}$	Immediate transition from the initial-state to the fail-dangerous state. Caused by dangerous failures in single components (e.g., single outputs) and by dangerous common cause failures.
$\lambda_{1,4} \dots \lambda_{1,n}$	Transition from the initial-state to an intermediate safe state. Caused by safe failures in redundant components (e.g., dual input modules). These failures can be safe detected, safe undetected and dangerous detected.
$\lambda_{1,n+1} \dots \lambda_{1,m}$	Transition from the initial-state to an intermediate safe state. Caused by dangerous undetected failures in redundant components (e.g., dual input modules).
$\lambda_{n+1,m+1} \dots \lambda_{m,2m-n}$	Transition from the intermediate safe state to an intermediate dangerous state. Caused by dangerous detected fault in a related redundant component (e.g., 1 output channel dangerous undetected combined with an output channel dangerous detected).
$\lambda_{4,2} \dots \lambda_{n,2}$	Transition from an intermediate safe state to the fail-safe state. Caused by a second safe fault in a redundant related component (e.g., dual input modules). These failures can be safe detected, safe undetected, dangerous detected and all failures that result in a transition from state 1 to state 2.
$\lambda_{4,3} \dots \lambda_{n,3}$	Transition from an intermediate safe state to the fail-dangerous state. Caused by the same failures that result in a transition from state 1 to state 3.
$\lambda_{n+1,2} \dots \lambda_{m,2}$	Transition from an intermediate safe state to the fail-safe state. Caused by all failures that result in a transition from state 1 to state 2.
$\lambda_{n+1,3} \dots \lambda_{m,3}$	Transition from an intermediate safe state to the fail-dangerous state. Caused by a second dangerous undetected fault in a redundant related component (e.g., 2 input modules) and all failures that result in a transition from state 1 to state 3.

Numerous other transitions are possible (e.g. from state m+1 to states 2 and 3, from state 4 to state m+1, etc.) but these are not drawn for clarity and as their contribution is negligible.

The matrix development methodology discussed in Annex A does not include the state transitions $\lambda_{4,2} \dots \lambda_{n,2}$, $\lambda_{n+1,2} \dots \lambda_{m,2}$, $\lambda_{4,3} \dots \lambda_{n,3}$, $\lambda_{n+1,3} \dots \lambda_{m,3}$ in the resulting Markov models as these transitions prevent the use of a closed form solution.

5 Procedures for quantification of logic solver performance

This clause will outline steps that will allow a vendor and user to agree on the information necessary to quantify and to document SIF component performance. The components include sensors, logic solvers, and final elements. This text illustrates procedures for logic solvers.

5.1 Assumptions and limitations

This clause will outline communications and agreements necessary to ensure proper modeling.

5.1.1 Assumptions and limitations inherent to the logic solver

The vendor should state:

- What data (e.g., functional test interval) is used for failure rates of cards or components, and what is the source (historical records and population size, or calculated and how) of the data.
- What data is used for diagnostic coverage of cards or components, and what is the source of the data.
- Common cause assumptions and the basis for the assumption.

5.1.2 Assumptions and limitations associated with the end user and the specific application

Since the logic solver PFD_{avg} is affected by the off-line functional test interval, the user should provide the intended test interval to the vendor for use in the calculations. This information may be generic to user company standards and practices, or may be dictated by the specific requirements of the application being modeled.

The following information needs to be available for all calculations:

- Power system failure rates

NOTE Some parts of the power supply may be outside the vendor's scope of supply. The interface must be defined.
- Redundancy of each component, communication channel, or card included in the calculation
- Time interval for off-line testing
- Time interval for off-line repair
- Time interval for on-line repair (this will require up-front analysis by the user and supplier for practicality and safety)

For PFD_{avg} calculation:

- Definition of each safety instrumented function and associated SIL
- Number of inputs and outputs for each SIF and redundancy of each.
(NOTE Inputs and outputs with no SIL requirement such as indicator lights or alarms can generally be omitted from the calculation.)
- Generally, the safety function case with the highest SIL requirement will be the case that determines whether the logic solver meets the performance requirements. When multiple safety function cases have the same SIL, select the case with the largest number of I/O, number of I/O channels, etc. to determine whether the logic solver meets the performance requirements.

For $MTTF^{spurious}$ calculations:

- Desired I/O groupings for spurious trip calculations
- Number of inputs and outputs for each group and redundancy of each.

(NOTE Number of I/O associated with spurious trip may be different than number associated with PFD_{avg} . I/O not associated with spurious trips should be omitted.)

5.1.3 Assumptions and limitations inherent in the quantification technique

The vendor should state:

- The type and source of the program used to perform calculations
- Any limitations and assumptions that may be inherent in the specific program used for the calculation
- Mission time that is assumed for the calculation

Some examples of limitations that might be inherent in modeling programs (see Clause 4.3 also):

- Failure rates are assumed to be constant.
- Inspection and repair is assumed to be perfect.

The vendor and user should reach a mutual understanding of how these limitations and assumptions relate to the users application, and whether the model is adequate for the intended use.

5.2 Calculations and reports

5.2.1 The vendor should incorporate all data and assumptions specific to the application in a model and perform the calculations.

5.2.2 Reports

Information for each calculation should include the following:

- The vendor and specific equipment being modeled
- The end user and the specific application(s) being modeled
- All assumptions and limitations (input data)
- Calculation methods (or specific models, programs, options)
- Who performed the calculations
- Date the calculations were performed
- PFD_{avg} and $MTTF^{spurious}$ for each case calculated, and any data specific to that case
- Any additional information specifically agreed to by the parties

6 Logic solver Markov models calculation results

The objective of this clause is to define the logic solver architectures that are selected as examples for the application of Markov modeling technique, define their associated Markov models and give the calculation results.

6.1 Description and results of the reliability calculation for three E/E/PE logic solver configurations including input data tables

6.1.1 Configuration names

Config 1 Single PE logic solver with single I/O, One-out-of-One (1oo1) shutdown logic

Config 2 Dual PE logic solver with dual I/O, One-out-of-Two (1oo2) shutdown logic

Config 3 Single E/E logic unit with dual Inputs and Outputs and 1oo2 voting logic implemented by either:

- relay, or
- solid state, or
- fail-safe solid state logic solvers

6.1.2 Overview of the three calculated configurations

The calculation results per configuration includes

- hardware failures;
- common cause failures; and
- systematic failures and diagnostic coverage factors.

Configurations are without sensors and final elements.

For each of the configurations the results consist of two graphs:

A: Probability of Fail-Dangerous (PFD) is related to the safety integrity level (SIL) as defined by ANSI/ISA-84.01-1996, IEC 61508, and IEC 61511. The PFD is typically plotted as function of the functional test interval and its relationship to other calculation variables can be more thoroughly understood by performing uncertainty analysis. The probability of fail-dangerous is derived from the **fail-dangerous state**.

B: Probability of Fail-Safe (PFS) is also typically plotted as a function of the functional test interval. The Probability of **fail-safe** is derived from the logic solver **fail-safe state (spurious trip)**.

The included uncertainty analysis shows three curves in each graph and indicates the 10th - the 50th - and the 90th percentile graph. The theory behind the uncertainty analysis is explained in Clause 5.9 of ISA-TR84.00.02-2002 – Part 1.

NOTE The 90th percentile graph indicates the line where 90% of the logic solvers have a lower Probability or 90% are better in terms of Safety Integrity or Spurious failures.

For Configuration 2 there are two types of additional graphs

C: Sensitivity graphs for the PFD and PFS.

D: Correlation graphs for the PFD and PFS.

NOTE The theory behind the sensitivity and correlation analysis is explained in Clause 5.9 of ISA-TR84.00.02-2002 – Part 1.

Configuration No. 3 is calculated for Relays, Solid State logic and Inherently fail-safe solid state logic.

6.1.3 Abbreviations used in the fail-dangerous sensitivity graphs

1oo1 (single) = One-out-of-One

1oo2 (serial) = One-out-of-Two

And gate = Dual Input voter

Coverage factory test = Coverage factor for the factory acceptance test (see tables)

DDfrac = Diagnostic coverage factor for dangerous failures

E = Electric logic solver

E = Electronic logic solver

Inp = Input

Outp = Output

PE = Programmable Electronic

PFD (Safety Integrity Level) = Fail-dangerous = Probability of failure on demand

PFS (Spurious Trip) = Fail-safe = Probability of fail-safe

Pow. sup = Power supply

Proc = Processor

S frac = Failure mode ratio safe – Unsafe failures

SD frac = Diagnostic coverage factors safe failures

Start Prob = Systematic failures

6.1.4 Input data tables used for the comparison calculations of the different E/E/PE logic solver configurations

The data used for the calculations are shown in the tables of Annex B.

6.2 Configuration drawings, Markov diagrams and calculation results

This clause describes the architecture models, Markov models and gives the graphic results for Probability of Failure on Demand (PFD) and the Probability to Fail-Safe (PFS).

6.2.1 Single PE with single I/O, one-out-of-one (1oo1) shutdown logic

Figure 6.1 shows the block diagram for the first architecture. It should be noted that the logic solver has only one main processor. For clarity one input and one output module are shown in Figure 6.1.

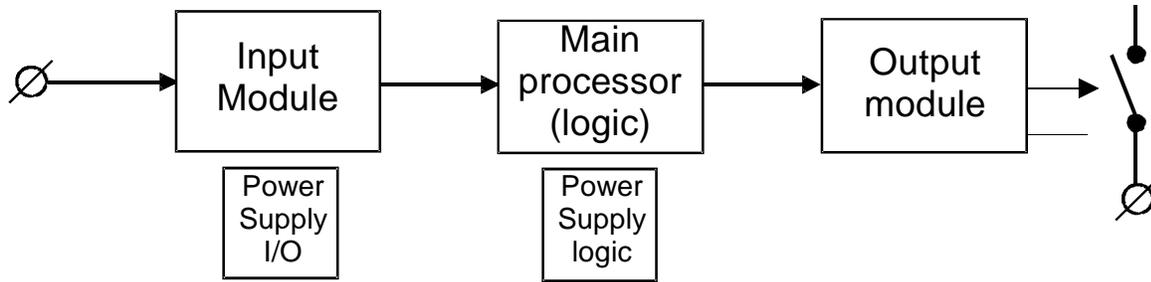


Figure 6.1 — Single PE logic solver with single I/O, one-out-of-one (1oo1) shutdown logic

The Markov model given in Figure 6.2 assumes there are *n* input modules and *m* output modules.

As mentioned previously, the safe and dangerous failure rates for each module in the logic solver are computed and then the dangerous detected and dangerous undetected failure rates are computed using the diagnostic coverage factors for dangerous failures ($C_{IM}^D, C_{IC}^D, C_{MP}^D, C_{OM}^D$, and C_{oc}^D).

The total safe and fail-dangerous failure rates for this simplex logic solver are the sum of the failure rates leading from state 1 to state 2 and from state 1 to state 3 respectively. As there is no redundancy in this architecture there are no intermediate states.

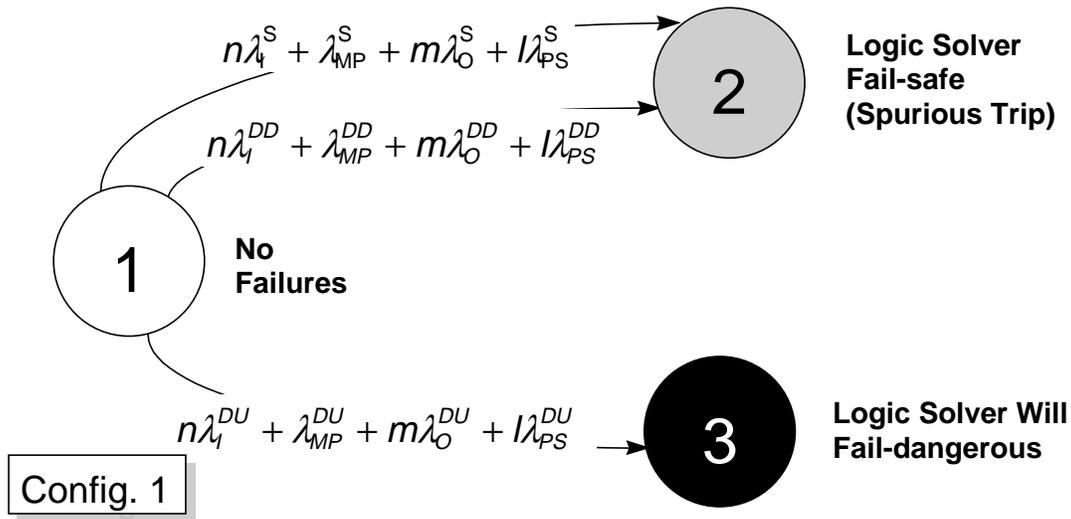


Figure 6.2 — Markov model Configuration 1: Single PE logic solver with single I/O, one-out-of-one (1oo1) shutdown logic

The graph provided in Figure 6.3 allows determination of probability of **fail-dangerous** of a single logic solver (1oo1) with a single input and output.

NOTE The 90th percentile graph indicates the line where 90% of the logic are better in terms of PFD.

The graph provided in Figure 6.4 allows determination of probability of fail-safe of a single logic solver (1oo1) with a single input and output.

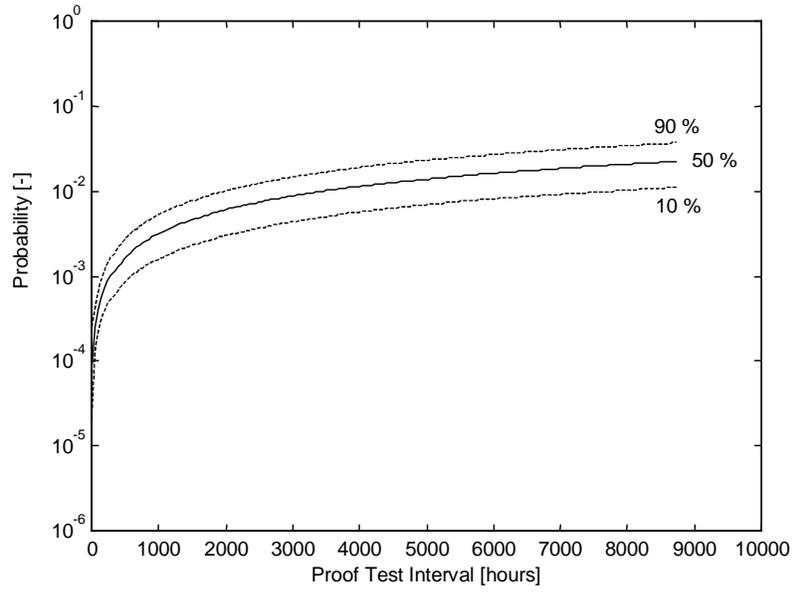
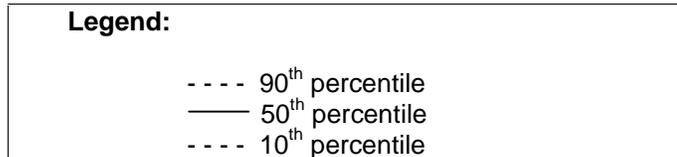


Figure 6.3 — PFD Configuration 1 – Probability fail-dangerous of a single PE logic solver with single input and output – 1oo1 configuration



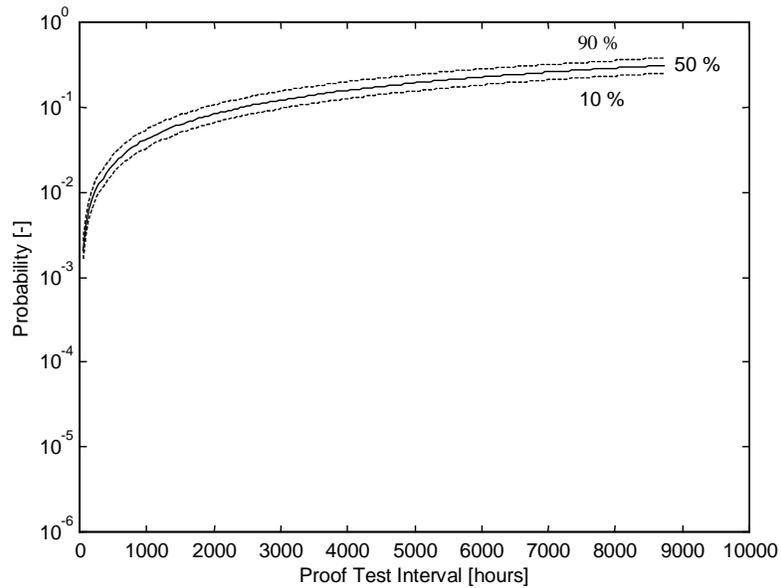


Figure 6.4 — PFS Configuration 1 – Probability of fail-safe of a single PE logic solver with single input and output – 1oo1 configuration

Legend:

- 90th percentile
- 50th percentile
- 10th percentile

6.2.2 Dual PE with dual I/O, one-out-of-two (1oo2) shutdown logic

A block diagram of this logic solver architecture is shown in Figure 6.5. This architecture consists of two completely independent legs. Each leg consists of a main processor with its associated I/O modules and power supplies. There is no communication between the two legs. The PE has two separate inputs, one input on each of the two legs or channels of the logic solver. Each output circuit consists of an output from each leg wired in series. Hence, each leg can independently open the output circuit and put the logic solver in the safe state.

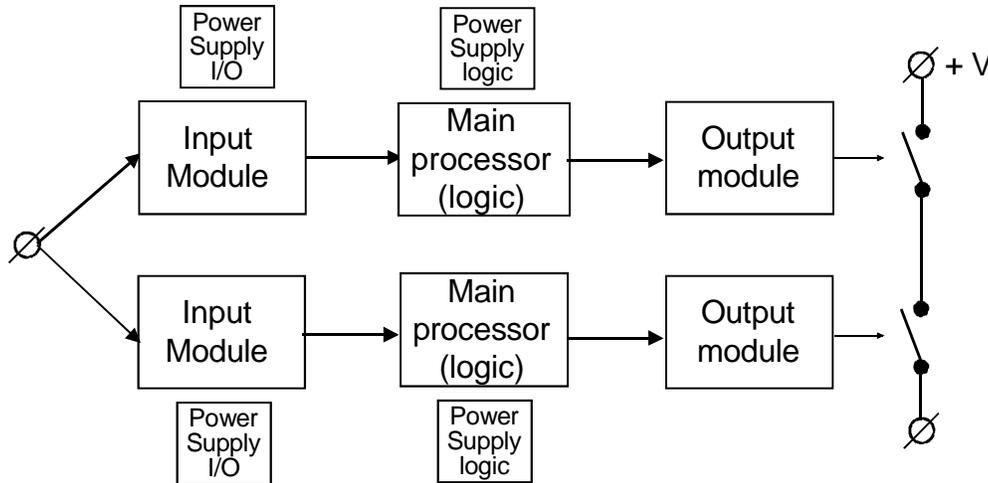


Figure 6.5 — Configuration 2 – Dual PE logic solver with dual I/O, one-out-of-two (1oo2) shutdown logic

The Markov model for this architecture is shown in Figure 6.6. The top arc in the Markov model represents a safe failure of any element in either leg of the PE. Any safe failure in either leg or channel will cause a spurious trip, because of the 1oo2 shutdown logic that allows either channel to shutdown the installation being protected. The bottom arc in the Markov model is for the common cause dangerous hardware failures.

Because of the dual redundancy and the 1oo2 logic, two failures of elements, one in each leg of the logic solver must occur before a fail-dangerous condition occurs. State 3 represents the state where the logic solver is in the fail-dangerous state and will not be repaired on-line since the failures are undetected. The PFD for the logic solver is computed by determining the probability of the logic solver being in state 3.

No inter-processor communication in this dual logic solver architecture is assumed. Since there is no inter-processor communication, each leg of the logic solver independently reads its inputs and determines what the outputs should be by execution of the shutdown logic IP (i.e., unable to respond to a demand). Note that a dangerous undetected failure of an input processor on one leg can result in a number of outputs on the leg stuck in the dangerous state. This is referenced in state 3 of Figure 6.6. So, the process will be in a fail-dangerous state if one of the corresponding outputs on the other leg fails in a dangerous state.

The states 4 to 9 are the safe intermediate states where a safe or detected failure has occurred that may be repaired before a subsequent failure brings the logic solver in the fail-safe or fail-dangerous state.

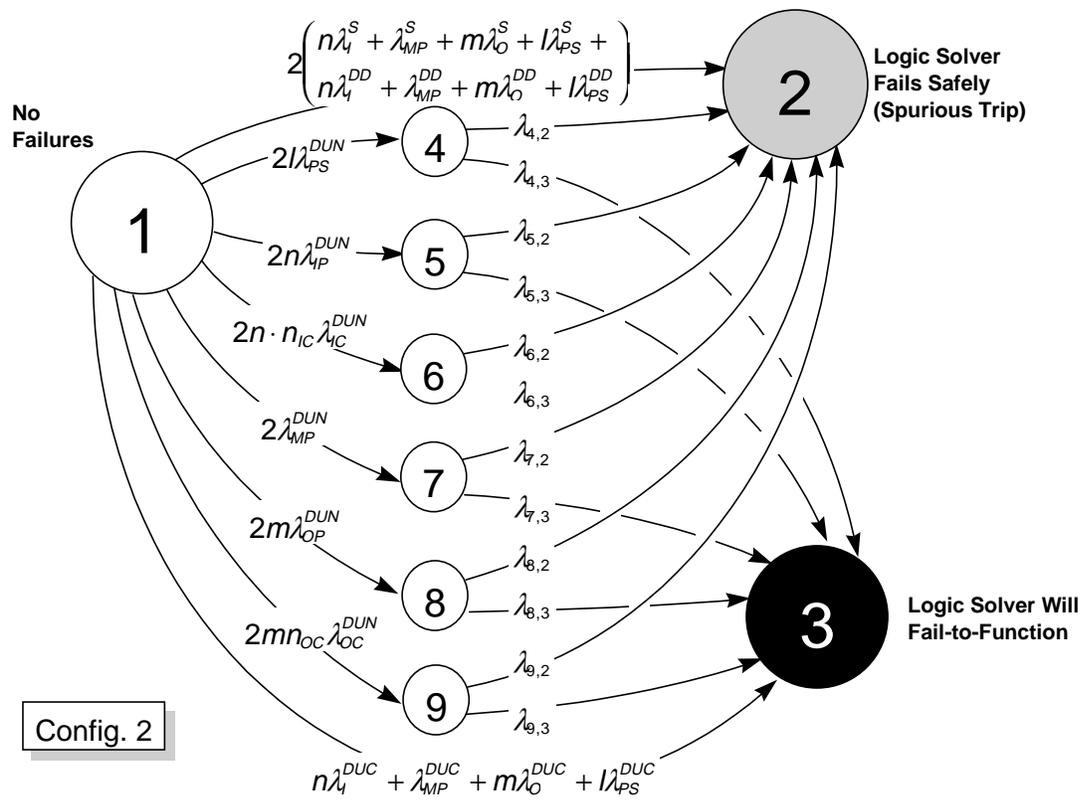


Figure 6.6 — Markov model for dual PE logic solver with dual I/O, one-out-of-two (1oo2) shutdown logic

The Markov model shown in 6.6, uses the following expressions of different failure rates denoted by the following λ 's:

$$\lambda_{4,2} = \lambda_{5,2} = \lambda_{6,2} = \lambda_{7,2} = \lambda_{8,2} = \lambda_{9,2} = 2(n\lambda_I^S + \lambda_{MP}^S + m\lambda_O^S + l\lambda_{PS}^S + n\lambda_I^{DD} + \lambda_{MP}^{DD} + m\lambda_O^{DD} + l\lambda_{PS}^{DD})$$

$$\lambda_{4,3} = n/I \lambda_I^{DU} + \lambda_{MP}^{DU} + m/I \lambda_O^{DU} + \lambda_{PS}^{DU}$$

$$\lambda_{5,3} = \lambda_I^{DU} + \lambda_{MP}^{DU} + f_{IPO} \lambda_O^{DU} + \lambda_{PS}^{DU}$$

$$\lambda_{6,3} = \lambda_{IP}^{DU} + \lambda_{IC}^{DU} + \lambda_{MP}^{DU} + f_{ICO} \lambda_O^{DU} + \lambda_{PS}^{DU}$$

$$\lambda_{7,3} = n\lambda_I^{DU} + \lambda_{MP}^{DU} + m\lambda_O^{DU} + l\lambda_{PS}^{DU}$$

$$\lambda_{8,3} = f_{OPI} \lambda_I^{DU} + \lambda_{MP}^{DU} + \lambda_O^{DU} + \lambda_{PS}^{DU}$$

$$\lambda_{9,3} = f_{OCI} \lambda_I^{DU} + \lambda_{MP}^{DU} + \lambda_{OP}^{DU} + \lambda_{OC}^{DU} + \lambda_{PS}^{DU}$$

$$\lambda^{DUN} = (1 - \beta)\lambda^{DU} = (1 - \beta)C^D \lambda^D$$

$$\lambda^{DUC} = \beta\lambda^{DU} = \beta C^{D\lambda D}$$

Config. 2

$$\lambda_{4,2} = \lambda_{5,2} = \lambda_{6,2} = \lambda_{7,2} = \lambda_{8,2} = \lambda_{9,2} = 2(n\lambda_I^S + \lambda_{MP}^S + m\lambda_O^S + l\lambda_{PS}^S + n\lambda_I^{DD})$$

Figure 6.7 — Transitions for the Markov model for the dual PE logic solver with dual I/O, one-out-of-two (1oo2) shutdown logic

The graph in Figure 6.8 allows determination of probability of fail-dangerous (PFD) of a dual logic solver (1oo2) with dual inputs and outputs.

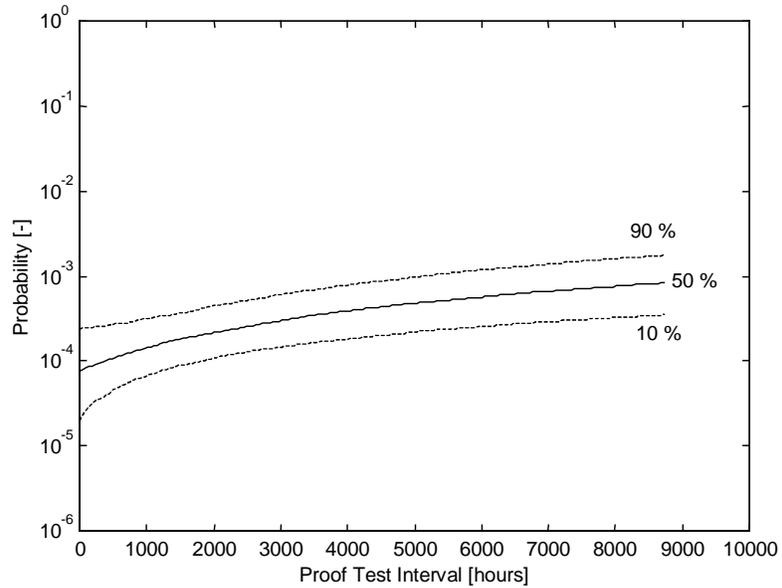


Figure 6.8 — PFD configuration 2 -- Probability fail-dangerous of a dual PE logic solver with dual inputs and outputs — 1oo2 configuration

Legend:

- 90th percentile
- 50th percentile
- 10th percentile

The bar chart in Figure 6.9 is a sensitivity graph of the probability calculation **fail-dangerous** dual logic solver (1oo2) with dual inputs and outputs. The theory behind the uncertainty plot is explained in ISA-TR84.00.02-2002 - Part 1, Clause 5.9.

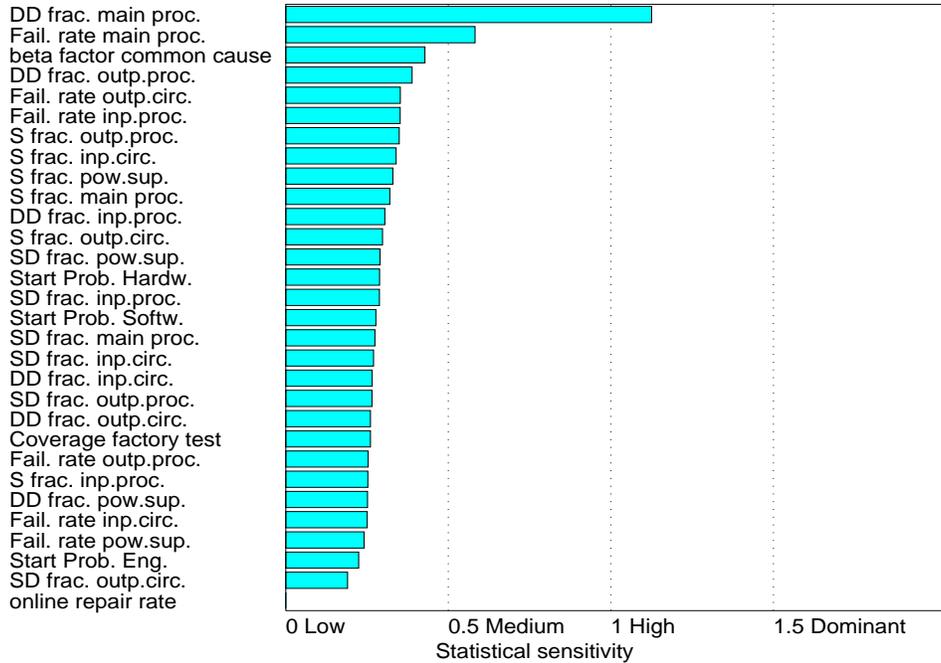


Figure 6.9 — Sensitivity PFD configuration 2 — Sensitivity graph of the probability calculation *fail-dangerous* — Dual PE logic solver with dual inputs and outputs — 1oo2 configuration

The bar chart in Figure 6.10 is a correlation graph of the probability calculation fail-dangerous logic solver (1oo2) with dual inputs and outputs.

NOTE The 90th percentile graph indicates the line where 90% of the logic are better in times of PFD.

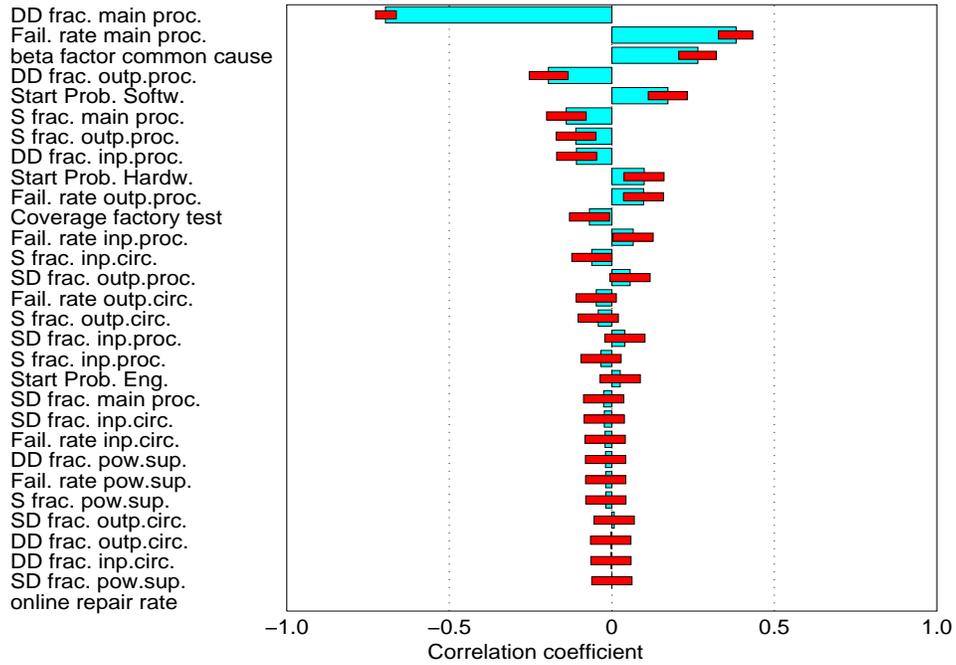


Figure 6.10 — Correlation PFD configuration 2 — Correlation graph of the probability calculation fail-dangerous — Dual PE logic solver with dual inputs and outputs — 1oo2 configuration

The graph in Figure 6.11 allows determination of probability of fail-safe of a dual logic solver (1oo2) with dual inputs and outputs.

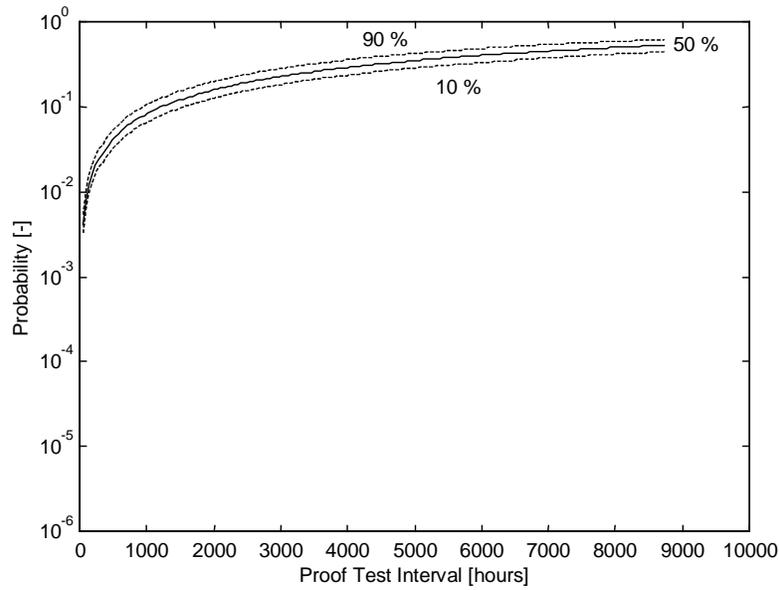
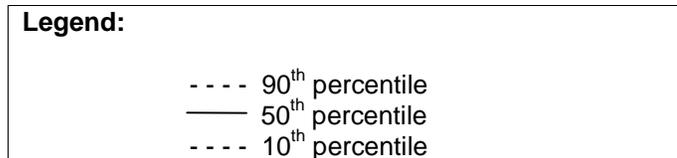


Figure 6.11 — PFS configuration 2 — Probability fail-safe of a dual PE logic solver with dual inputs and outputs — 1oo2 configuration



The bar chart in Figure 6.12 is a sensitivity graph of the probability calculation of fail-safe dual logic solver (1oo2) with dual inputs and outputs.

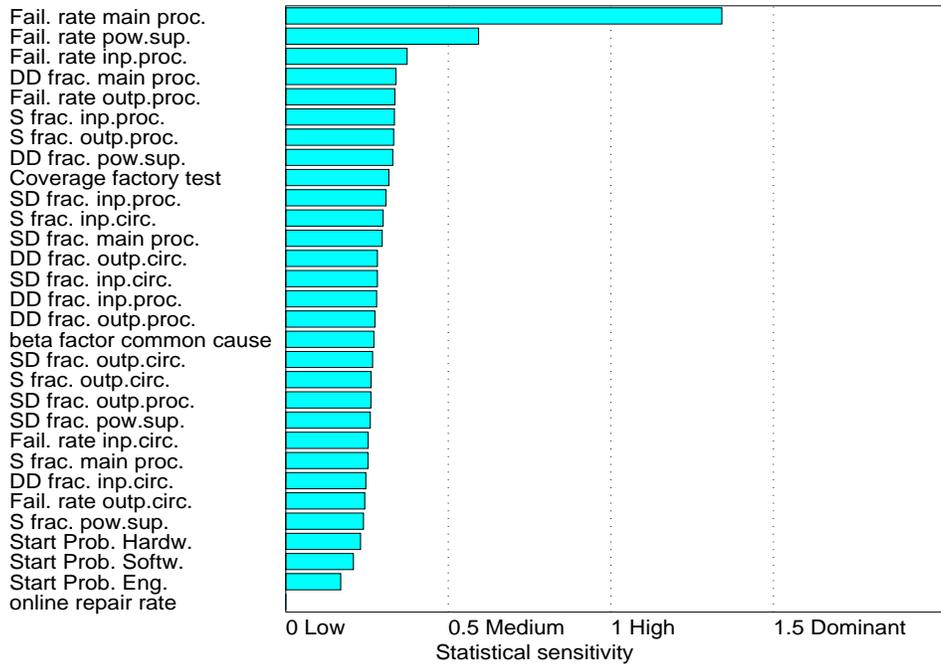


Figure 6.12 — Sensitivity PFS configuration 2 — Sensitivity graph of the probability calculation fail-safe — Dual PE logic solver with dual inputs and outputs — 1oo2 configuration

The bar chart in Figure 6.13 is a correlation graph of the probability calculation of fail-safe dual logic solver (1oo2) with dual inputs and outputs.

NOTE The 90th percentile graph indicates the line where 90% of the logic are better in times of PFD.

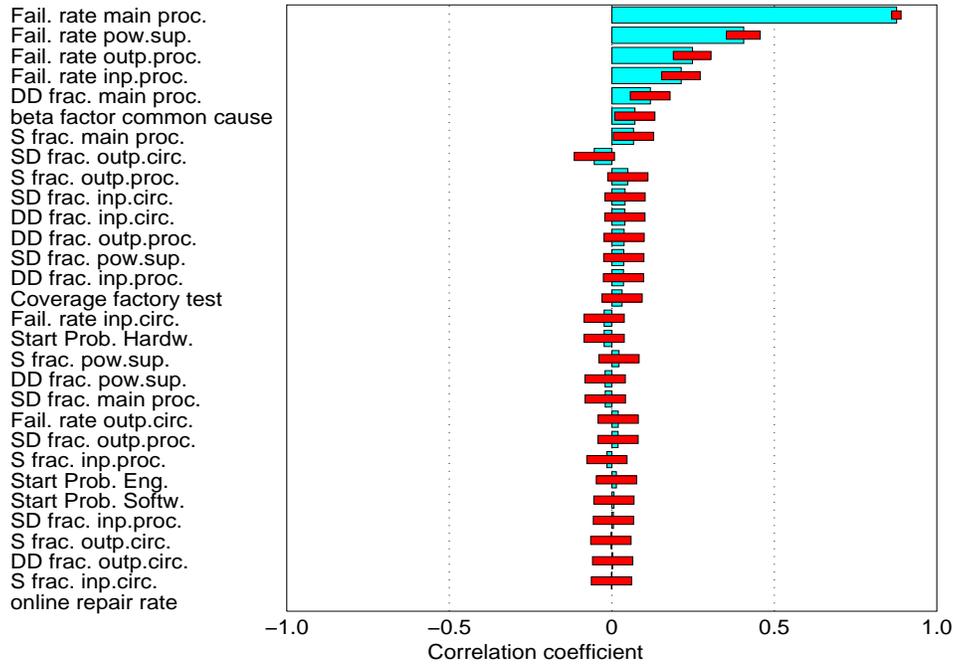


Figure 6.13 — Correlation PFS configuration 2 — Correlation graph of the probability calculation fail-safe — Dual PE logic solver with dual inputs and outputs — 1oo2 configuration

6.2.3 Single electrical/electronic logic solver with dual inputs and dual outputs

Figure 6.14 shows the block diagram of this architecture that has a hardware input voter which allows input data for the two input legs to be voted (1oo2) before performing the logic operations. Outputs are voted by connecting the output contacts or final elements in series (1oo2). Three typical aspects of this architecture are:

- a) The logic solver has an input voter that allows all inputs to be voted (1oo2). Hence, an input failure on one input channel does not propagate to the output and vice versa a faulty output does not influence the voting on the inputs.
- b) The logic solver in this architecture is single (electrical or electronic). The logic solver technologies can include:
 - 1) Relay
 - 2) Solid state
 - 3) Fail-safe solid state

- c) A redundant set of power supplies is assumed for this architecture, making it single fault tolerant for safe or detected power supply failures.

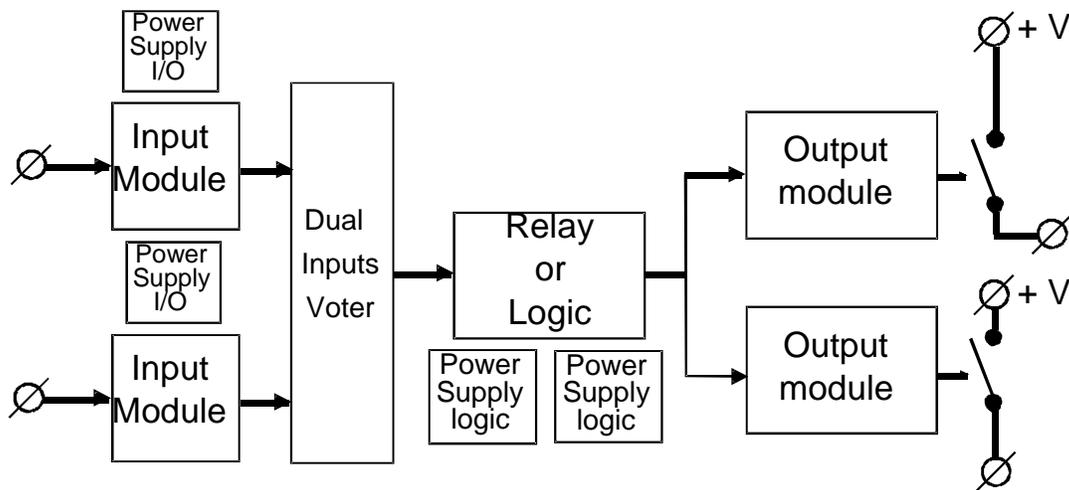


Figure 6.14 — Configuration 3 — Single E/E logic solver with dual inputs and outputs, one-out-of-two (1oo2) input voting logic

The Markov model for this architecture is shown in Figure 6.15. The top arc in the Markov model represents the safe or detected failures in the logic solver plus the common cause safe or detected failure rate of any element in either leg of the redundant components. The bottom arc in the Markov model represents the dangerous undetected failures in the logic solver plus the common cause dangerous undetected failure rate of any element in either redundant component. The states 4 and 5 are safe intermediate states where a safe or detected failure has occurred in any element of the logic solver that may be repaired before a subsequent failure brings the logic solver in the fail-safe or **fail-dangerous** state. The states 8 and 9 are the dangerous intermediate states where a combination of a dangerous undetected failure followed by a safe or dangerous detected failure resulted in a **fail-dangerous** state from which the logic solver may return to one of the states 6 or 7 by repairing the safe or dangerous detected failure.

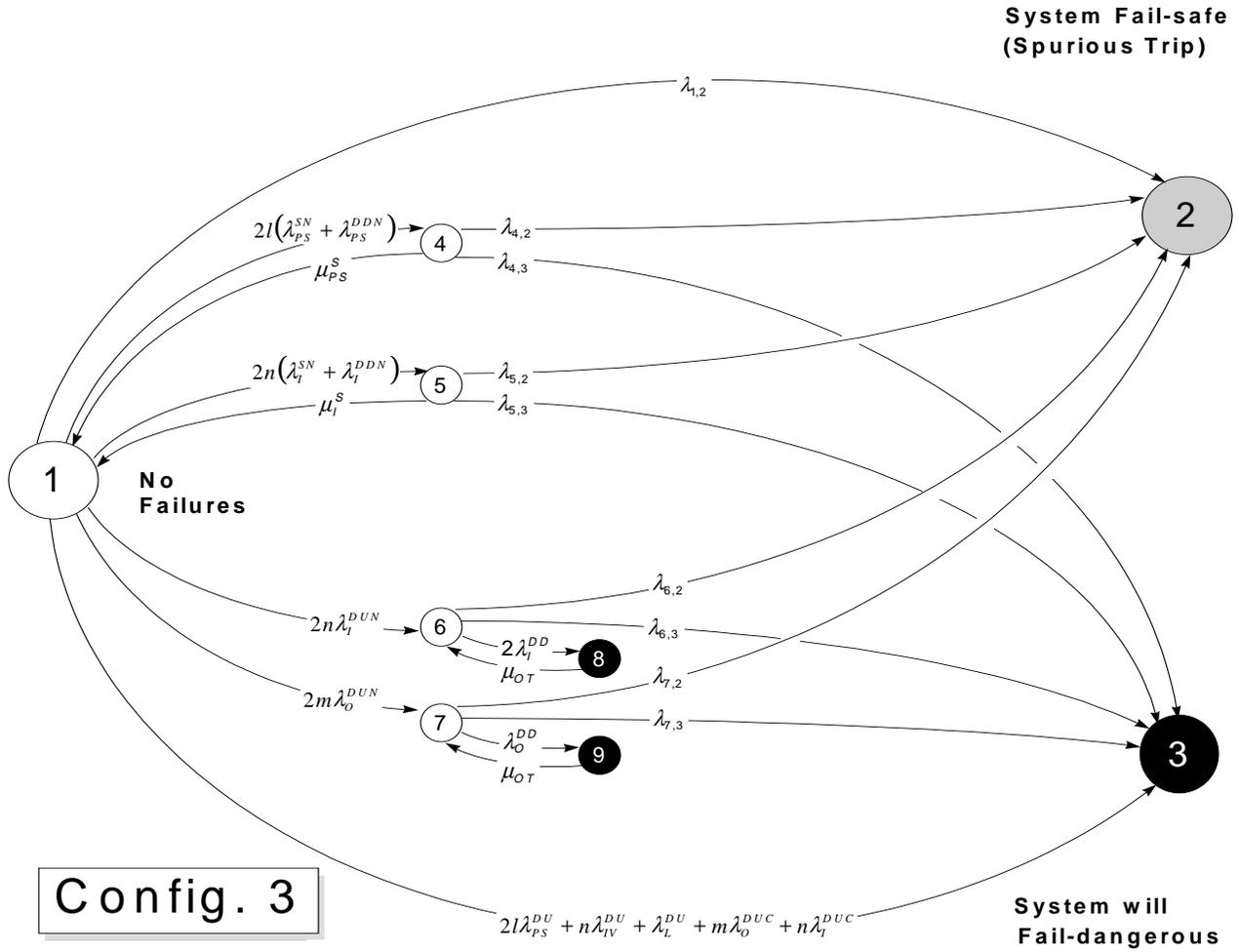


Figure 6.15 — Markov model for single E/E logic solver with dual inputs and outputs, one-out-of-two (1oo2) input voting logic

The Markov model shown in 6.15, uses the following expressions (Figure 6.16) of different failure rates denoted by the following λ s:

$$\begin{aligned}\lambda_{1,2} &= n(\lambda_{IV}^S + \lambda_{IV}^{DD}) + \lambda_L^S + \lambda_L^{DD} + 2m(\lambda_O^S + \lambda_O^{DD}) + l(\lambda_{PS}^{SC} + \lambda_{PS}^{DDC}) + n(\lambda_I^{SC} + \lambda_I^{DDC}) \\ \lambda_{4,2} &= \lambda_{PS}^S + \lambda_{PS}^{DD} + n(\lambda_{IV}^S + \lambda_{IV}^{DD}) + \lambda_L^S + \lambda_L^{DD} + 2m(\lambda_O^S + \lambda_O^{DD}) + n(\lambda_I^{SC} + \lambda_I^{DDC}) \\ \lambda_{5,2} &= 2(\lambda_I^S + \lambda_I^{DD}) + n(\lambda_{IV}^S + \lambda_{IV}^{DD}) + \lambda_L^S + \lambda_L^{DD} + 2m(\lambda_O^S + \lambda_O^{DD}) + l(\lambda_{PS}^{SC} + \lambda_{PS}^{DDC}) + (n-1)(\lambda_I^{SC} + \lambda_I^{DDC}) \\ \lambda_{6,2} &= \lambda_{7,2} = \lambda_{1,2} \\ \lambda_{4,3} &= \lambda_{5,3} = 2l\lambda_{PS}^{DU} + n\lambda_{IV}^{DU} + \lambda_L^{DU} + m\lambda_O^{DUC} + n\lambda_I^{DUC} \\ \lambda_{6,3} &= 2(l\lambda_{PS}^{DU} + \lambda_I^{DU}) + n\lambda_{IV}^{DU} + \lambda_L^{DU} + m\lambda_O^{DUC} + (n-1)(\lambda_I^{DUC}) \\ \lambda_{7,3} &= 2l\lambda_{PS}^{DU} + \lambda_O^{DU} + n\lambda_{IV}^{DU} + \lambda_L^{DU} + (m-1)\lambda_O^{DUC} + n\lambda_I^{DUC}\end{aligned}$$

Config. 3

Figure 6.16 — Transitions for Markov model for single E/E logic solver with dual inputs and outputs, one-out-of-two (1oo2) input voting logic

$$\begin{aligned}\lambda_{1,2} &= n(\lambda_{IV}^S + \lambda_{IV}^{DD}) + \lambda_L^S + \lambda_L^{DD} + 2m(\lambda_O^S + \lambda_O^{DD}) + l(\lambda_{PS}^{SC} + \lambda_{PS}^{DDC}) + n(\lambda_I^{SC} + \lambda_I^{DDC}) \\ \lambda_{4,2} &= \lambda_{PS}^S + \lambda_{PS}^{DD} + n(\lambda_{IV}^S + \lambda_{IV}^{DD}) + \lambda_L^S + \lambda_L^{DD} + 2m(\lambda_O^S + \lambda_O^{DD}) + n(\lambda_I^{SC} + \lambda_I^{DDC}) \\ \lambda_{5,2} &= 2(\lambda_I^S + \lambda_I^{DD}) + n(\lambda_{IV}^S + \lambda_{IV}^{DD}) + \lambda_L^S + \lambda_L^{DD} + 2m(\lambda_O^S + \lambda_O^{DD}) + l(\lambda_{PS}^{SC} + \lambda_{PS}^{DDC}) + (n-1)(\lambda_I^{SC} + \lambda_I^{DDC}) \\ \lambda_{6,2} &= \lambda_{7,2} = \lambda_{1,2} \\ \lambda_{4,3} &= \lambda_{5,3} = 2l\lambda_{PS}^{DU} + n\lambda_{IV}^{DU} + \lambda_L^{DU} + m\lambda_O^{DUC} + n\lambda_I^{DUC} \\ \lambda_{6,3} &= 2(l\lambda_{PS}^{DU} + \lambda_I^{DU}) + n\lambda_{IV}^{DU} + \lambda_L^{DU} + m\lambda_O^{DUC} + (n-1)(\lambda_I^{DUC}) \\ \lambda_{7,3} &= 2l\lambda_{PS}^{DU} + \lambda_O^{DU} + n\lambda_{IV}^{DU} + \lambda_L^{DU} + (m-1)\lambda_O^{DUC} + n\lambda_I^{DUC}\end{aligned}$$

Config. 3

Figure 6.17 — PFD configuration 3 — (1) Relay — Probability of *fail-dangerous* of a single E (relay) logic solver with dual inputs and outputs — 1oo2 input voting logic

The graph in Figure 6.18 allows determination of probability of fail-safe of a single relay logic solver (1oo2) with dual inputs and outputs.

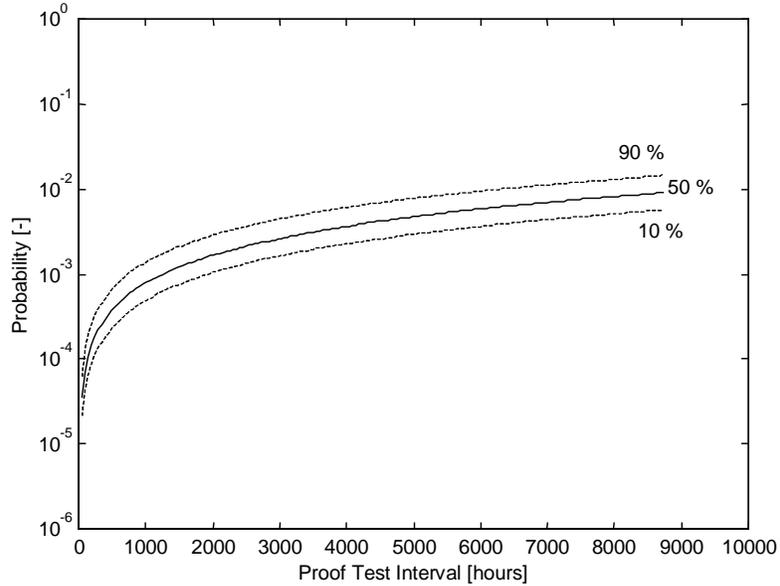
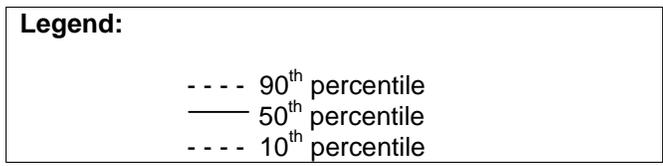


Figure 6.18 — PFS configuration 3 — (1) relay — Probability of fail-safe of a single E (relay) logic solver with dual inputs and outputs – 1oo2 input voting logic



The graph in Figure 6.19 allows determination of probability of **fail-dangerous** of a single solid state logic solver (1oo2) with dual inputs and outputs.

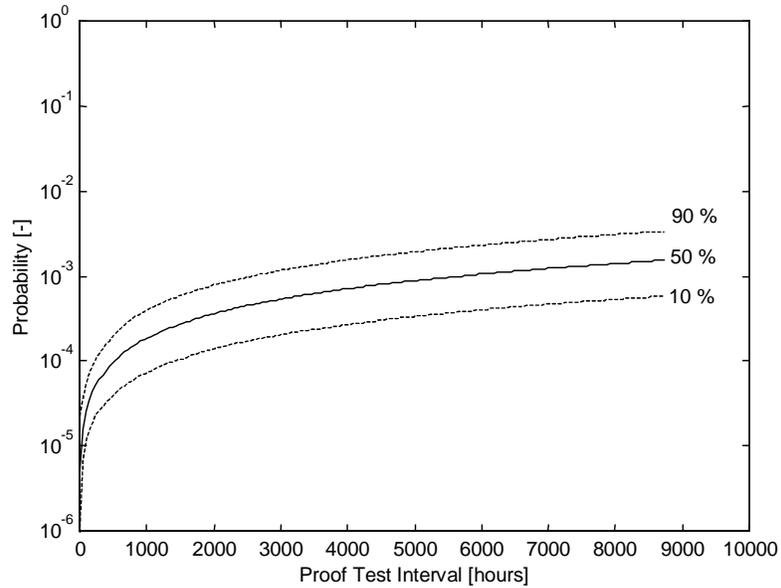


Figure 6.19 — PFD configuration 3 — (2) solid State — Probability of fail-dangerous of a single E (solid state) logic solver with dual inputs and outputs — 1oo2 input voting logic

Legend:

- 90th percentile
- 50th percentile
- 10th percentile

The graph in Figure 6.20 allows determination of probability of fail-safe of a single solid state logic solver (1oo2) with dual inputs and outputs.

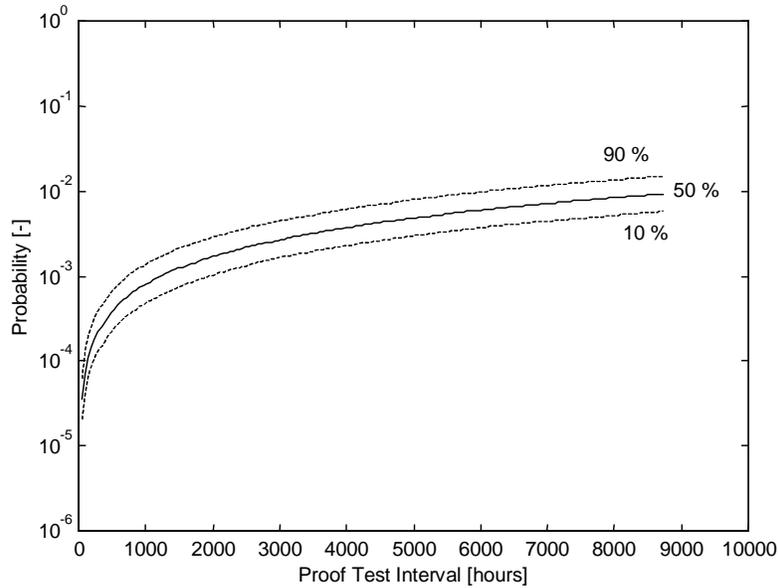
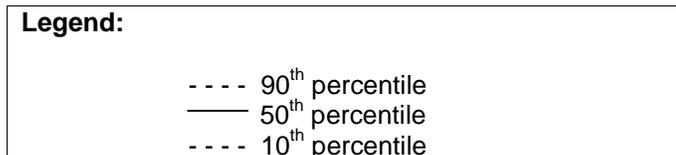


Figure 6.20 — PFS configuration 3 — (2) solid state — Probability of fail-safe of a single E (solid state) logic solver with dual inputs and outputs — 1oo2 input voting logic



The graph in Figure 6.21 allows determination of probability of **fail-dangerous** of a single fail-safe solid state logic solver (1oo2) with dual inputs and outputs.

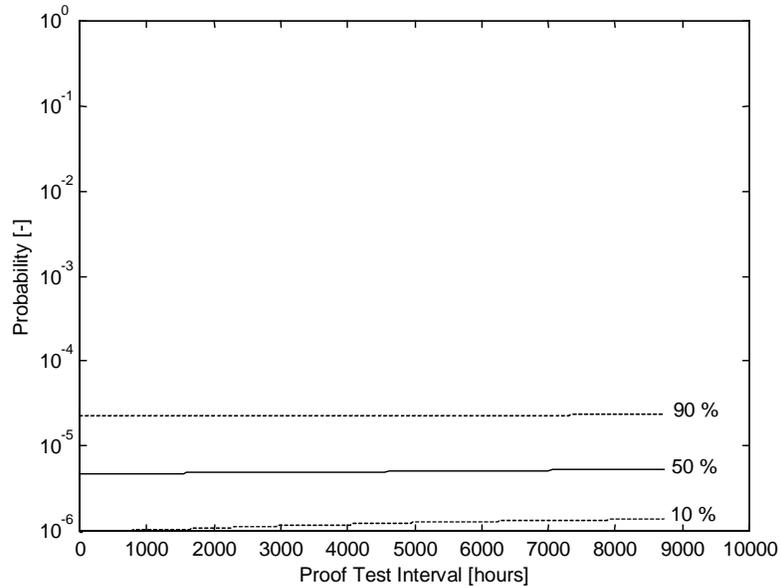


Figure 6.21 — PFD configuration 3 — (3) fail-safe solid state — Probability of fail-dangerous of a single E (fail-safe solid state) logic solver with dual inputs and outputs — 1oo2 input voting logic

Legend:

- 90th percentile
- 50th percentile
- 10th percentile

NOTE Because of the inherent features in fail-safe logic, the major reason this system can fail-dangerous is because of systematic failure. Figure 6.21 is modeled such that

- systematic failures occur at t=0; and
- the remaining time its probability to fail-dangerous is assumed to be very small.

The result is the graphs having very flat plots.

The graph in Figure 6.22 allows determination of probability of fail-safe of a single fail-safe solid state logic solver (1oo2) with dual inputs and outputs.

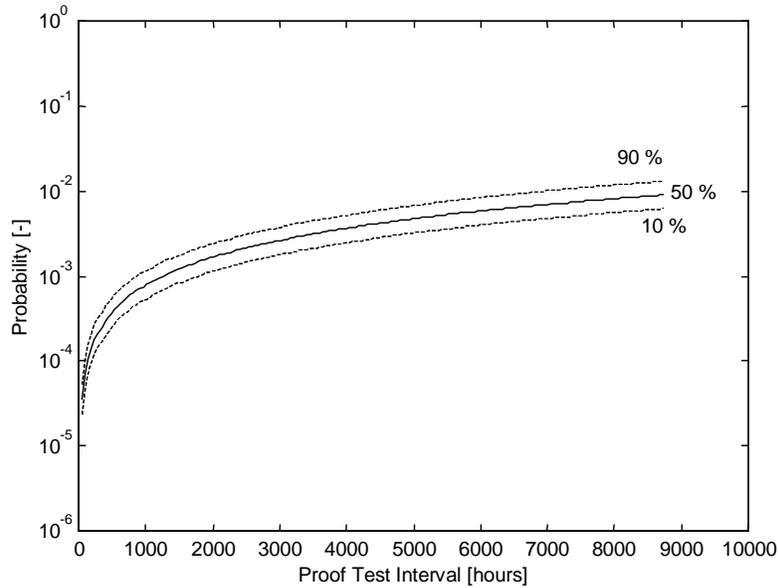
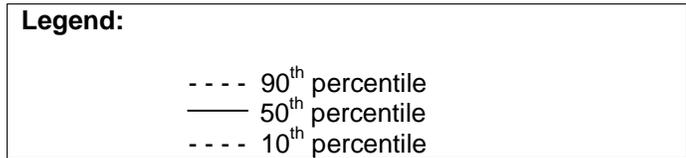


Figure 6.22 — PFS configuration 3 — (3) fail-safe solid state — Probability of fail-safe of a single E (fail-safe solid state) logic solver with dual inputs and outputs — 1oo2 input voting logic



This page intentionally left blank.

Annex A (informative) — Markov model development and quantification

A.1 Introduction

This annex describes how to develop Markov Models for a safety instrumented system (SIS) logic solver where both safety and availability are important. Markov Models are developed for both fail-safe and **fail-dangerous** states. The **fail-dangerous** Markov Model is used to determine the **PFDavg** (average probability of failure on demand). The Fail-Safe Markov Model is used to determine **MTTF^{spurious}** (mean time to a spurious trip) or **PFS** (probability of failing safe). The annex also describes how these models can be simplified, so **PFDavg** and **MTTF^{spurious}** can be calculated in a simple manner.

A.2 Model development

Markov models are created by identifying all the possible states that the system may enter while transitioning from a fully operational logic solver, through partially failed (degraded) states, to a failed system. To accomplish this task, the system states are identified during the FMEA and the corresponding transition probabilities (i.e., probabilities of components that must fail in order to transition from one state to another state) are shown as arcs on the Markov model.

Markov model construction starts with the state of the system where all of the components are functioning properly (successful state). To develop the other system states, the following general procedure is followed: For any system state, list all of the operating components and the ways the system may leave that state. There are two ways: a) a successful component in the state fails or; b) failed component in the state is repaired or replaced. In the former case, the probability of a component failure is the driving mechanism to force a transition out of the state. For exponential failure and repair probability distributions and using the rare event approximation (Annex A, Clause A.4, Equation 3) the probability of failure is defined as λt , where λ is the failure rate of the component and t is the time. For the latter case, the repair probability is given as μt , where μ is the repair rate. Due to convention, these probabilities in the Markov models are shown as simply failure rates and/or repair rates and are commonly referred to as transition rates. The transition probabilities are always considered in the formulation and analysis of the models.

To illustrate how a Markov model is created, we will examine the PE logic solver shown in Figure A.1 which is a Dual PE logic solver having Dual Input / Output modules, with 1oo2 shutdown logic. The results of the FMEA are used to create Table A.1. The component failure categories that are considered for the fail-dangerous Markov Model are printed bold in Table A.1.

Table A.1 — Component failure categories

Component	Failure Categories					
	SCC	DCC	SD	SU	DD	DU
IP	SCC	DCC	SD	SU	DD	DU
OP	SCC	DCC	SD	SU	DD	DU
MP	SCC	DCC	SD	SU	DD	DU
IC	SCC	DCC	SD	SU	DD	DU
OC	SCC	DCC	SD	SU	DD	DU

Legend: Component

- IP – Common part of input module
- OP – Common part of output module
- MP – Main Processor
- IC – Input Channel
- OC – Output Channel

Failure Categories

- SCC – Safe common cause failure
- DCC – Dangerous common cause failure
- SD – Safe detected hardware failure
- SU – Safe undetected hardware failure
- DD – Dangerous detected hardware failure
- DU – Dangerous undetected hardware failure

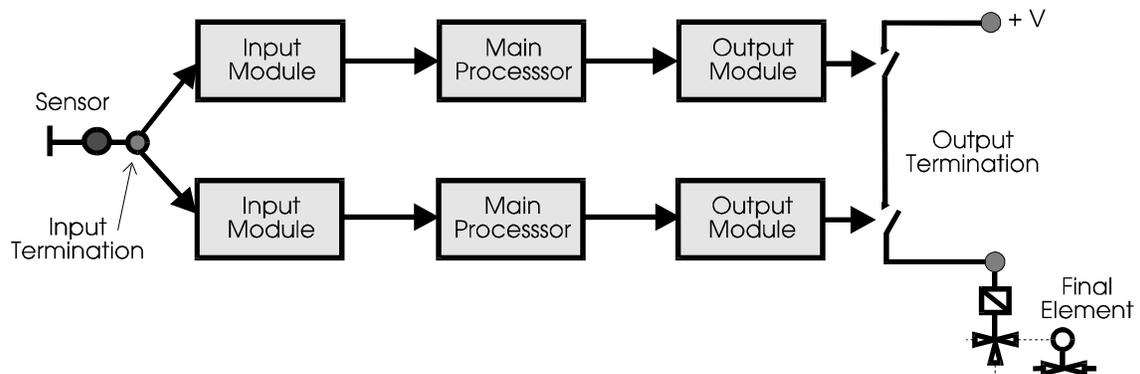


Figure A.1 — Dual programmable (PE) logic solver with dual I/O, 1oo2 shutdown logic

The first state, state 1, is the PE logic solver success state where all the components are functioning properly. One safe failure of any of the components will transition the logic solver into a failed state

because of the output configuration of the PE logic solver (series). The component failure categories that are considered for the fail spurious Markov Model are printed bold in Table A.2.

Table A.2 — Component failure categories for fail spurious Markov models

Component	Failure Categories					
IP	SCC	DCC	SD	SU	DD	DU
OP	SCC	DCC	SD	SU	DD	DU
MP	SCC	DCC	SD	SU	DD	DU
IC	SCC	DCC	SD	SU	DD	DU
OC	SCC	DCC	SD	SU	DD	DU

Legend: Component
 IP – Common part of input module
 OP – Common part of output module
 MP – Main Processor
 IC – Input Channel
 OC – Output Channel

Failure Categories
 SCC – Safe common cause failure
 DCC – Dangerous common cause failure
 SD – Safe detected hardware failure
 SU – Safe undetected hardware failure
 DD – Dangerous detected hardware failure
 DU – Dangerous undetected hardware failure

These component failures have identified the failed safe state. The interim Fail Spurious Markov model is shown in Figure A.2.

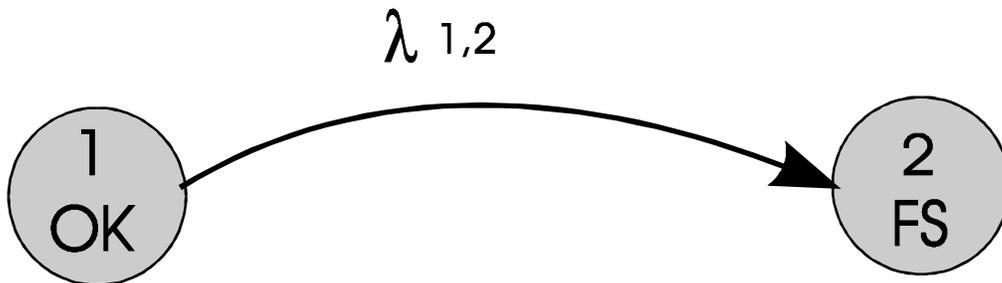


Figure A.2 — Interim Fail Spurious Markov model

Since any of the components failing in either of the two modes (safe detected/undetected) will force the PE logic solver into the fail spurious state, the total failure probability to leave state 1 and to fail to state 2 is:

$$(Eq. A.1) \quad P_{1,2} = (\lambda_{cc}^s + \lambda_f^s + \lambda_{ip}^s + \lambda_{op}^s + \lambda_{mp}^s + \lambda_{ic}^s + \lambda_{oc}^s)t = \lambda_1 t$$

The total failure rate from state 1 to state 2 is:

$$(Eq. A.2) \quad \lambda_1 = p_2 \lambda_{cc}^s + \lambda_f^s + 2[n \lambda_i^s + \lambda_{mp}^s + m \lambda_o^s]$$

where,

$$(Eq. A.3) \quad \lambda_i^s = \lambda_{ip}^s + n_{ic} \lambda_{ic}^s$$

$$(Eq. A.4) \quad \lambda_o^s = \lambda_{op}^s + n_{oc} \lambda_{oc}^s$$

$$(Eq. A.5) \quad \lambda_{ps}^s = \lambda_{ps}^{sd} + \lambda_{ps}^{su}$$

$$(Eq. A.6) \quad \lambda_a^s = \lambda_a^{sd} + \lambda_a^{su}$$

The rest of the failure categories shown in Table A.1 must be modeled in the same manner. Since two dangerous failures are required to bring the PE logic solver to the failed dangerous state, there should be some intermediate states that the PE logic solver will transition into when only one dangerous failure occurs. In addition, these intermediate states should be partitioned between the dangerous detected (DD) and the dangerous undetected (DU) categories. The partition is made to account for the different inspection and repair actions for components that fail in these two modes. Dangerous detected failures are detected by the PE logic solver and are repaired with a rate μ_{ot} . Dangerous undetected failures are detected during a scheduled inspection and are repaired with a rate μ_{pt} . The interim model with the intermediate states for dangerous undetected failures is shown in Figure A.3. Only the dangerous undetected states are shown to avoid the creation of a complex figure. Nevertheless, the states corresponding to the dangerous detected failures are analogous to the states shown in Figure A.3.

The transition rates for the intermediate dangerous undetected states are:

$$(Eq. A.7) \quad \lambda_{1,3} = 2n \lambda_{ip}^{du} \lambda_{3,1} = \mu_{pt}$$

$$(Eq. A.8) \quad \lambda_{1,4} = 2m \lambda_{op}^{du} \lambda_{4,1} = \mu_{pt}$$

(Eq. A.9) $\lambda_{1,5} = 2 \lambda_{mp}^{du} \lambda_{5,1} = \mu_{pt}$

(Eq. A.10) $\lambda_{1,6} = 2n * n_{ic} \lambda_{ic}^{du} \lambda_{6,1} = \mu_{pt}$

(Eq. A.11) $\lambda_{1,7} = 2m * n_{oc} \lambda_{oc}^{du} \lambda_{7,1} = \mu_{pt}$

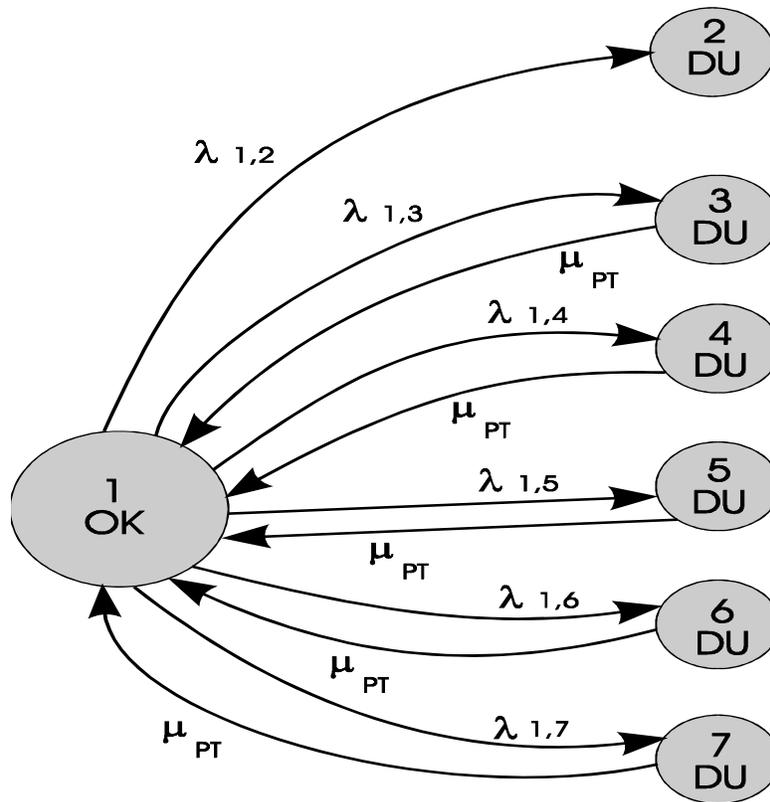


Figure A.3 — Markov Model for dangerous detected failures

The states that correspond to dangerous detected failures would be modeled in a similar way. They are 5 states, namely states 8, 9, 10, 11 and 12. These states are not shown on Figure A.3 due to space restrictions. The transition rates for these states are obtained by changing the failure rate superscript from du to dd and the periodic repair rate μ_{pt} with the on line repair rate μ_{ot} in Equations 7 through 11. For example, state 8 corresponds to a dangerous detected failure of an input processor. The transition rates to state 8 from state 1 and the repair from state 8 to state 1 are:

$$(Eq. A.12) \quad \lambda_{1,8} = 2n \lambda_{ip}^{dd} \quad \lambda_{8,1} = \mu_{ot}$$

From these intermediate states, states 3 through 12, transitions may occur to bring the logic solver to the failed dangerous state, which is state 13. These transitions account for the second dangerous failure mentioned earlier in the text. The question that must be asked at every intermediate state is: **Given that the logic solver is in an intermediate state, say state 3, what operating component of that state must fail in order for the logic solver to reach state 13?** The answer to this question defines the transition rate from state 3 to state 13.

For example, state 3 has been defined by a dangerous undetected failure of an input microprocessor in one of the PE logic solvers. In order for the logic solver to fail to state 13 one of the following must occur:

1. The corresponding input processor on the working PE logic solver fails in a dangerous undetected mode.
2. The corresponding input channels on the working PE logic solver must fail in a dangerous undetected mode.
3. The remaining (functioning) main processor must fail in a dangerous undetected mode.
4. The output processor on the working PE logic solver associated with the same input signals that the first failed input processor was handling must fail in a dangerous undetected mode.
5. The output channels on the working PE logic solver associated with the same input signals that the first failed input processor was handling must fail in a dangerous undetected mode.

Since the logic solver has 2 input and output modules, items 1 and 2 above clearly refer to the dangerous undetected failure of the remaining input module. Similarly, items 4 and 5 describe the failure of the output channels of the remaining module that are associated with the original failed input channels. Therefore, the transition rates between state 3 and state 13 are:

$$(Eq. A.13) \quad \lambda_{3,13} = \lambda_{ip}^{du} + n_{ic} \lambda_{ic}^{du} + \lambda_{mp}^{du} + f_{ipo} [\lambda_{op}^{du} + n_{oc} \lambda_{oc}^{du}] \quad \lambda_{13,1} = \mu_{pt}$$

The transition from state 3 to state 13 is shown on Figure A.4.

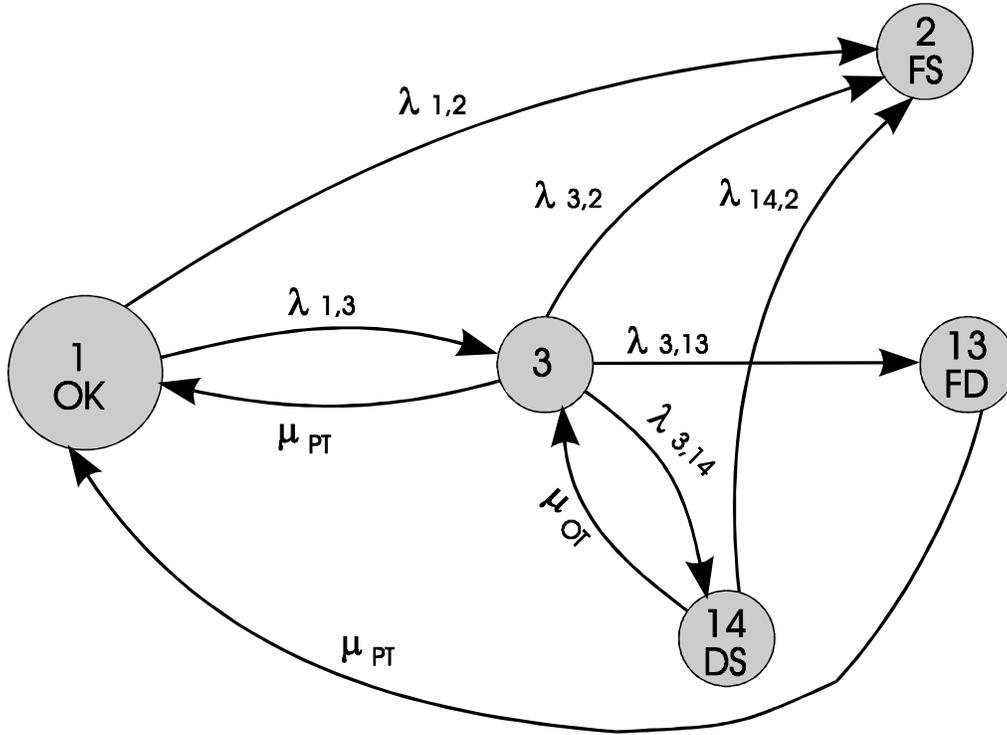


Figure A.4 — Markov Model for transition from state 3 to state 13

Similar relations can be obtained for the transition rate from the other dangerous undetected failure states to state 13. The transition rates are the following:

$$(Eq. A.14) \quad \lambda_{4,13} = f_{opi} [\lambda_{ip}^{du} + n_{ic} \lambda_{ic}^{du}] + \lambda_{mp}^{du} + \lambda_{op}^{du} + n_{oc} \lambda_{oc}^{du} \quad \lambda_{13,4} = \mu_{pt}$$

$$(Eq. A.15) \quad \lambda_{5,13} = n [\lambda_{ip}^{du} + n_{ic} \lambda_{ic}^{du}] + \lambda_{mp}^{du} + m [\lambda_{op}^{du} + n_{oc} \lambda_{oc}^{du}] \quad \lambda_{13,5} = \mu_{pt}$$

$$(Eq. A.16) \quad \lambda_{6,13} = \lambda_{ip}^{du} + n_{ic} \lambda_{ic}^{du} + \lambda_{mp}^{du} + f_{ico} [\lambda_{op}^{du} + n_{oc} \lambda_{oc}^{du}] \quad \lambda_{13,6} = \mu_{pt}$$

$$(Eq. A.17) \quad \lambda_{7,13} = f_{oci} [\lambda_{ip}^{du} + n_{ic} \lambda_{ic}^{du}] + \lambda_{mp}^{du} + \lambda_{op}^{du} + n_{oc} \lambda_{oc}^{du} \quad \lambda_{13,7} = \mu_{pt}$$

The above rates (Equations 13-17) account only for a second dangerous undetected failure that will bring the PE logic solver to a failed state (state 13). It must be noted that other failures (e.g., dangerous detected, safe detected and safe undetected) can also occur to force the PE logic solver to other states which may not necessarily be state 13.

The modeling also must account for these failures. To illustrate this modeling step, assume that the logic solver is in state 3 (an input processor has failed in a dangerous undetected mode). We have accounted for the second dangerous undetected failure that will bring the logic solver to state 13 (see Equation 13 and Figure A.4). What happens if the PE logic solver in state 3 experiences a safe detected failure of one of the functioning components? The PE logic solver will transition to the fail spurious state (state 2) since a single safe detected failure will de-energize the output circuit. The same will occur if the PE logic solver in state 3 experiences a safe undetected failure. Now we have two additional transitions (failures) that must be included in the model as shown in Figure A.4.

Now consider the case of a second dangerous but detected failure from state 3 before the original failure is repaired. The PE logic solver will fail to another state. Obviously both output contacts are forced closed and the PE logic solver will not be able to respond if a demand occurs. Is the PE logic solver in a failed dangerous state (i.e., state 13)? In order to answer this question, examine the sequence of failures from the beginning. The PE logic solver experiences one dangerous undetected failure. The operator(s) does not know of this failure and assumes the logic solver is fully operational.

Now the PE logic solver experiences a second dangerous but detected failure. The operator knows of this failure and will repair the logic solver. The repair action will eliminate the second dangerous detected failure but still leave the first undetected failure. Therefore, the PE logic solver is indeed in a failed dangerous state but not state 13 because the dangerous detected failure will be repaired. This new state, state 14, is a failed dangerous state only for the time necessary to repair the second dangerous detected failure. While the logic solver is in state 14 and the second dangerous detected failure is repaired the logic solver can fail to state 13 if a dangerous undetected failure occurs, or to state 2 if a dangerous detected failure occurs. To illustrate this modeling approach, Figure A.4 shows all the transitions that could occur from state 3.

The transition rates from state 3 to the states shown in Figure A.4 are:

$$(Eq. A.18) \quad \lambda_{3,14} = \lambda_{ip}^{dd} + n_{ic} \lambda_{ic}^{dd} + \lambda_{mp}^{dd} + f_{ipo} [\lambda_{op}^{dd} + n_{oc} \lambda_{oc}^{dd}] \quad \lambda_{14,3} = \mu_{ot}$$

$$(Eq. A.19) \quad \lambda_{3,2} = \lambda_{ip}^s + n_{ic} \lambda_{ic}^s + \lambda_{mp}^s + \lambda_{op}^s + n_{oc} \lambda_{oc}^s$$

The transition from state 3 to state 2, $\lambda_{3,2}$ is a valid transition. Examining the overall logic solver and how the logic solver will eventually fail from state 1 to state 2, there are several failure paths. The path that dominates is $\lambda_{1,2}$ because a single component failure is required. Any other path, such as $\lambda_{3,2}$ requires two failures. For example, the probability to have a transition directly from state 1 to state 2 is:

$$(Eq. A.20) \quad P_{1,2} = \lambda_1 t$$

where λ_1 is defined in Equation 2.

The probability for the logic solver to be found in state 2 having gone through transitions from state 1 and 3 is:

$$(Eq. A.21) \quad P_{1,2} = P_{1,3} * P_{3,2} = \lambda_{1,3} t * \lambda_{3,2} t$$

From Equations 20 and 21, it can be deduced that the probability to have the PE logic solver in state 2 is in fact dominated by the single transition λ_1 . The same reasoning determines that the transition from state 3 to state 13 is dominated by $\lambda_{3,13}$ which is much greater than the combination of transitions from state 13 to 14 and from 14 to 13, $\lambda_{3,14} * \lambda_{14,13}$.

The same procedure may be followed for the transition from the remainder of the dangerous undetected states (4, 5, 7 and 8) to the failed dangerous state (13), to the failed safe state (2) and to additional new states that are defined by a second dangerous detected failure. These new states are 15, 16, 17 and 18. The transition rates from states 4, 5, 6 and 7 to states 15, 16, 17 and 18 are:

$$(Eq. A.22) \quad \lambda_{4,15} = f_{op} I [\lambda_{ip}^{dd} + n_{ic} \lambda_{ic}^{dd}] + \lambda_{mp}^{dd} + \lambda_{op}^{dd} + n_{oc} \lambda_{oc}^{dd} \quad \lambda_{15,4} = \mu_{ot}$$

$$(Eq. A.23) \quad \lambda_{5,16} = n [\lambda_{ip}^{dd} + n_{ic} \lambda_{ic}^{dd}] + \lambda_{mp}^{dd} + m [\lambda_{op}^{dd} + n_{oc} \lambda_{oc}^{dd}] \quad \lambda_{16,5} = \mu_{ot}$$

$$(Eq. A.24) \quad \lambda_{6,17} = \lambda_{ip}^{dd} + \lambda_{ic}^{dd} + \lambda_{mp}^{dd} + f_{ico} [\lambda_{op}^{dd} + n_{oc} \lambda_{oc}^{dd}] \quad \lambda_{17,6} = \mu_{ot}$$

$$(Eq. A.25) \quad \lambda_{7,18} = f_{oci} [\lambda_{ip}^{dd} + n_{ic} \lambda_{ic}^{dd}] + \lambda_{mp}^{dd} + \lambda_{op}^{dd} + \lambda_{oc}^{dd} \quad \lambda_{18,7} = \mu_{ot}$$

The failure categories that remain to be considered are the secondary failures from states 8, 9, 10, 11 and 12 (one dangerous detected failure). The same approach can be taken as those for the secondary failures from states 3 through 7. Specifically, for every state (8-12) the logic solver can have all four types of secondary failures (SD, SU, DD and DU). To illustrate the methodology, examine state 8 and the secondary transitions from this state.

State 8 corresponds to a single dangerous detected failure of an input microprocessor. The failure is detected and will be repaired on line. In the time that it takes the failure to be repaired, a secondary failure may occur. This secondary failure may be:

1. A safe detected failure on the second (working) leg that will bring the logic solver to state 2.
2. A safe undetected failure that will bring the logic solver to state 2.
3. A dangerous detected failure on the working leg that will bring the logic solver to the fail spurious state (state 2) because the logic solver has now experience two dangerous detected failures.
4. A dangerous undetected failure on the working leg. This second failure will bring the logic solver to a new set of states analogous to states 14-18. These states will be failed dangerous states for the limited time that it takes to repair the first detected failure.

The transitions from state 8 to all other states are shown in Figure A.5. Similar transitions exist for states 9, 10, 11 and 12 which correspond to a dangerous detected failure of an output processor, a main processor, an input channel and an output channel, respectively. It is noted that a second dangerous undetected failure from any of the states (8-12) creates a new state that is a failed dangerous state. These new states, namely 19, 20, 21, 22 and 23, are distinguished from state 13 because the operator will repair the first logic solver detected failure and the logic solver will transition to the state corresponding to the remaining undetected dangerous failure. In fact, the logic solver will transition to one of the states 3, 4, 5, 6 or 7 (marked DU in Figure A.5) depending on the component that has failed dangerous undetected. To illustrate these concepts, examine the transitions from state 8. This state corresponds to a dangerous detected failure of an input microprocessor. All the transitions are shown in Figure A.5.

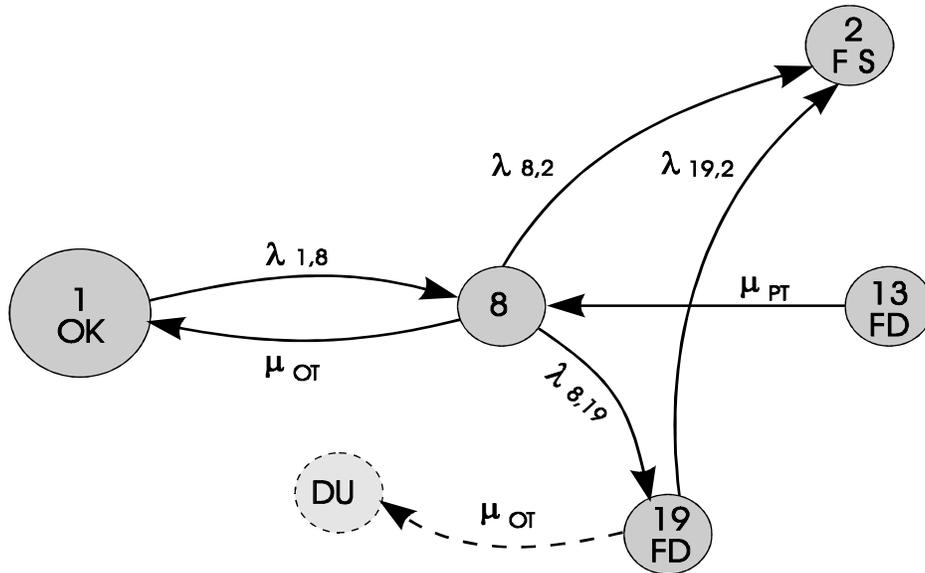


Figure A.5 — Markov Model for transition from state 8 to all other states

As was mentioned before, any secondary safe detected or undetected failure will bring the PE logic solver to the fail spurious state. In addition, the transition of the PE logic solver to the fail spurious state is dominated by λ_1 (see Equations 20 and 21). A second safe detected failure will also bring the PE logic solver to a fail spurious state. What is of concern is the second dangerous undetected failure from state 8. This failure can occur with a rate given below

$$(Eq. A.26) \quad \lambda_{8,19} = \lambda_{ip}^{du} + n_{ic} \lambda_{ic}^{du} + \lambda_{mp}^{du} + f_{ipo} [\lambda_{op}^{du} + n_{oc} \lambda_{oc}^{du}] \quad \lambda_{14,3} = \mu_{ot}$$

The rates from states 9, 10, 11 and 12 to the new states 20, 21, 22 and 23 are given below:

$$(Eq. A.27) \quad \lambda_{9,20} = f_{op} I [\lambda_{ip}^{du} + n_{ic} \lambda_{ic}^{du}] + \lambda_{mp}^{du} + \lambda_{op}^{du} + n_{oc} \lambda_{oc}^{du} \quad \lambda_{20,9} = \mu_{ot}$$

$$(Eq. A.28) \quad \lambda_{10,21} = n [\lambda_{ip}^{du} + n_{ic} \lambda_{ic}^{du}] + \lambda_{mp}^{du} + m [\lambda_{op}^{du} + n_{oc} \lambda_{oc}^{du}] \quad \lambda_{21,10} = \mu_{ot}$$

$$(Eq. A.29) \quad \lambda_{11,22} = \lambda_{ip}^{du} + \lambda_{ic}^{du} + \lambda_{mp}^{du} + f_{ico} [\lambda_{op}^{du} + n_{oc} \lambda_{oc}^{du}] \quad \lambda_{22,11} = \mu_{ot}$$

$$(Eq. A.30) \quad \lambda_{12,23} = f_{oci} [\lambda_{ip}^{du} + n_{ic} \lambda_{ic}^{du}] + \lambda_{mp}^{du} + \lambda_{op}^{du} + \lambda_{oc}^{du} \quad \lambda_{23,12} = \mu_{ot}$$

Notice that the transition out of state 19 is the on line repair rate. The logic solver had identified the first dangerous detected failure of the input microprocessor. While the logic solver was being repaired, the second dangerous undetected failure occurred to bring the logic solver into state 19. Therefore, state 19 is a fail-dangerous state during the time it takes to repair the logic solver. The repair of the logic solver refers to repairing the first dangerous detected failure which when completed leaves the logic solver with only one dangerous undetected failure. Thus the logic solver will not transition back to state 8. It will transition back to one of the states (3-7) that corresponds to the second dangerous undetected component failure. Which state it actually enters depends on the specific component in Equation 26 that actually failed. Therefore, it seems that each of the dangerous detected states, namely 8-12, should transition to five other states depending on the specific component that fails in a dangerous undetected mode in order to allow the logic solver to transition back to a specific dangerous undetected state (3-7) after the first dangerous detected failure has been repaired. To illustrate this concept, Figure A.6 shows all the states that the logic solver can transition into starting from state 8 and having a second dangerous undetected failure. These states, 24, 25, 26, 27 and 28 correspond to a second dangerous undetected failure of an input processor, an output processor, a main processor, an input and output channel, respectively.

The transitions from state 8 to these new states, shown on Figure A.6, are:

$$(Eq. A.31) \quad \lambda_{8,24} = \lambda_{ip}^{du} \quad \lambda_{24,3} = \mu_{ot}$$

$$(Eq. A.32) \quad \lambda_{8,25} = f_{ipo} \lambda_{op}^{du} \quad \lambda_{25,4} = \mu_{ot}$$

$$(Eq. A.33) \quad \lambda_{8,26} = \lambda_{mp}^{du} \quad \lambda_{26,5} = \mu_{ot}$$

$$(Eq. A.34) \quad \lambda_{8,27} = n_{ic} \lambda_{ic}^{du} \quad \lambda_{27,6} = \mu_{ot}$$

$$(Eq. A.35) \quad \lambda_{8,28} = f_{ipo} n_{oc} \lambda_{oc}^{du} \quad \lambda_{28,7} = \mu_{ot}$$

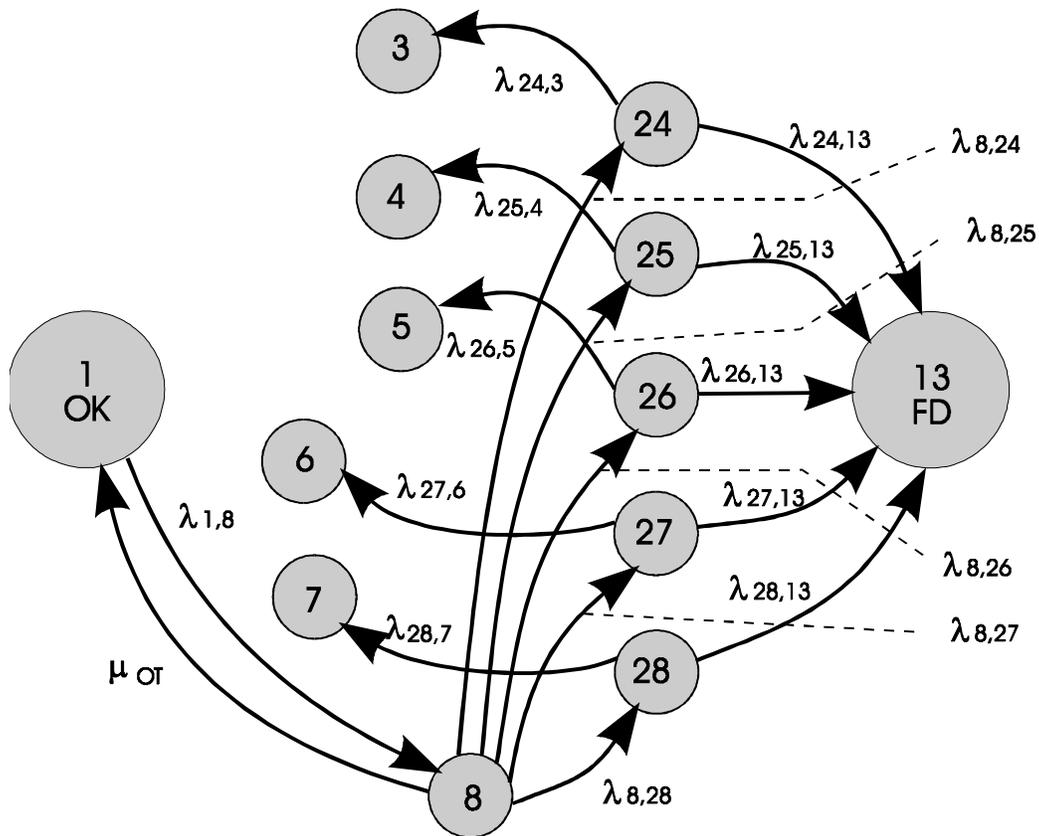


Figure A.6 — Markov Model for transition from state 8 to a second dangerous undetected failure

Equations 31-35 are simply the five parts of Equation 26. Similar relationships exist for the transitions from states 9-12 to new states (e.g., from state 9 the logic solver would transition to 5 different states). Comparing Figures A.5 and A.6, the only difference between them is that state 19 (Figure A.5) has been decomposed to the five states (24-28) to allow for the on-line repair of the dangerous detected failure of the input microprocessor and transition the logic solver back to a specific state (3-7) that depends on the specific component that failed in a dangerous undetected mode.

The same arguments can be made for the transitions from state 8 (or 9-12) to all the states shown on Figure A.6. Specifically, the transition of the logic solver from state 8 to state 2 is dominated by $\lambda_{1,2}$. The transition from state 8 to state 13 is small and can occur in two ways. First, transition from 8 to 24 and from 24 to 13. For example, transition from 8 to 13 is $\lambda_{8,13} = \lambda_{8,24} * \lambda_{24,13}$. The other way is to transition from 8 to 24, from 24 to 3 and from 3 to 13. Therefore, $\lambda_{8,13} = \lambda_{8,24} * \lambda_{24,3} * \lambda_{3,13}$.

From the above detailed analysis, the Markov model for the PE logic solver under investigation consists of more than 40 states. In this document, some assumptions have been made in an attempt to: a) simplify the Markov models to make them easier to quantify and still have the required accuracy and modeling detail; b) provide close form solutions for the simplified Markov models and; c) allow the development of simplified numerical solutions.

The assumptions made in order to simplify the Markov models are:

1. Failure rates and repair rates are assumed to be constant.
2. The time interval (mission time) that will be used to evaluate the reliability of a PE logic solver. It is assumed to be the time between periodic off-line test intervals of the logic solver. This assumption eliminates all of the periodic repair, μ_{pt} , in the Markov models. The effect of this assumption on the model is shown on Figures A.7 and A.8 which are the same as Figures A.3 and A.4, respectively, without the periodic repair transition.

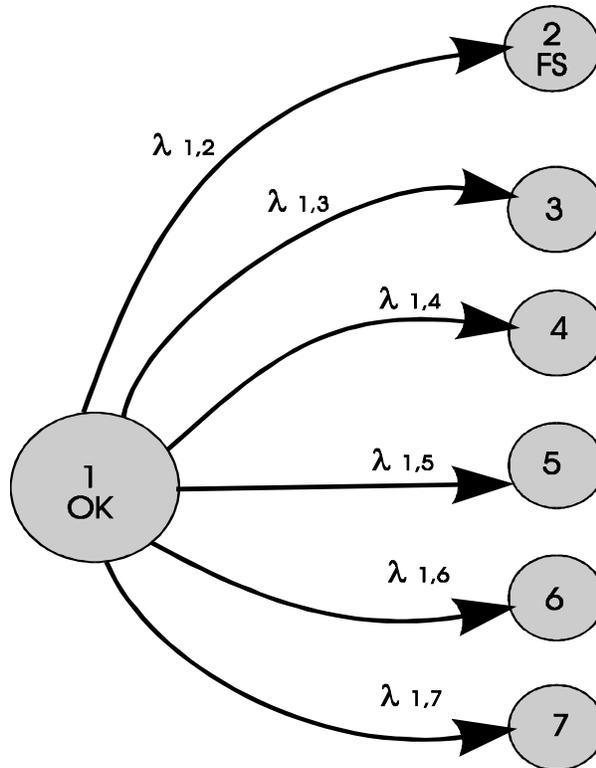


Figure A.7 — Markov model for dangerous undetected failures without the periodic repair transition

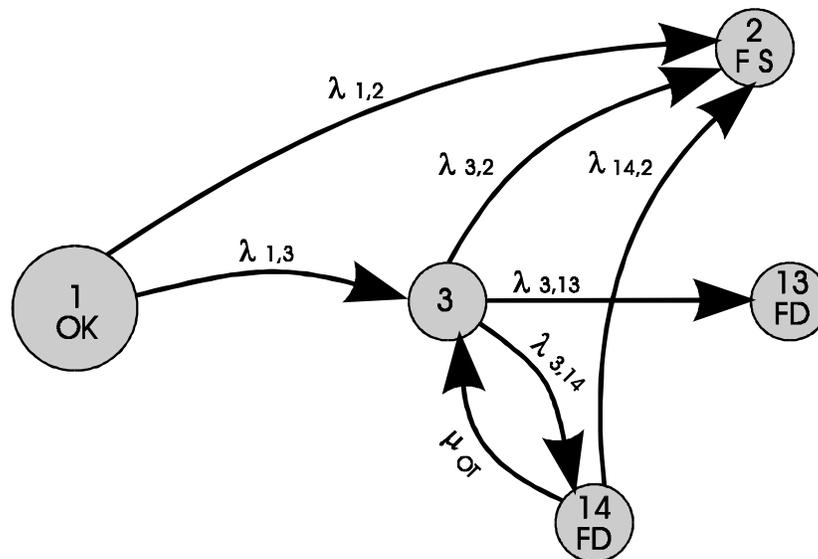


Figure A.8 — Markov model for transition from state 3 to state 13 without the periodic repair transition

1. Dangerous detected failures are neglected in fail to danger part of the Markov models. The effect on the models is to eliminate all the states and transitions that occur between state 1 and states 8-12 and all subsequent transitions from these states. Therefore, the states shown in Figures A.5 and A.6 have been neglected. The reasons that these states have been neglected are the following:
2. States 8-12: The logic solver transitions from state 1 to any of these states with a failure rate which is much smaller than the on-line repair rate used to bring the logic solver back to state 1. Hence, the logic solver will reside in states 8-12 for a very small time compared to the states 3-7 (i.e. the probability for the logic solver to follow the dangerous undetected failure path is much greater than the dangerous detected failure path).
3. Secondary transitions from states 8-12 to any other state: Given that the PE logic solver is in one of these states, it has been shown that it can transition to any number of states (Figures A.5 and A.6). The parameters of importance in any reliability analysis are the probability to have the logic solver in a fail spurious or fail-dangerous state. It has been shown that in order for the PE logic solver to transition from any of these states (8-12) to failed states it requires additional transitions. These additional transitions are negligible when compared to the transition rates λ_1 for the fail spurious state and transition rates from states 3-7 to state 13.
4. The transition rates from all states to the fail spurious state (2) and fail-dangerous state (13) have been examined. Similar arguments to those of Equations 20 and 21 have been used to allow the elimination of transitions from same states to the two failed states that are much smaller when compared to primary transitions rates. To illustrate the effect of these assumptions, Figure A.4 has been recreated as Figure A.9.

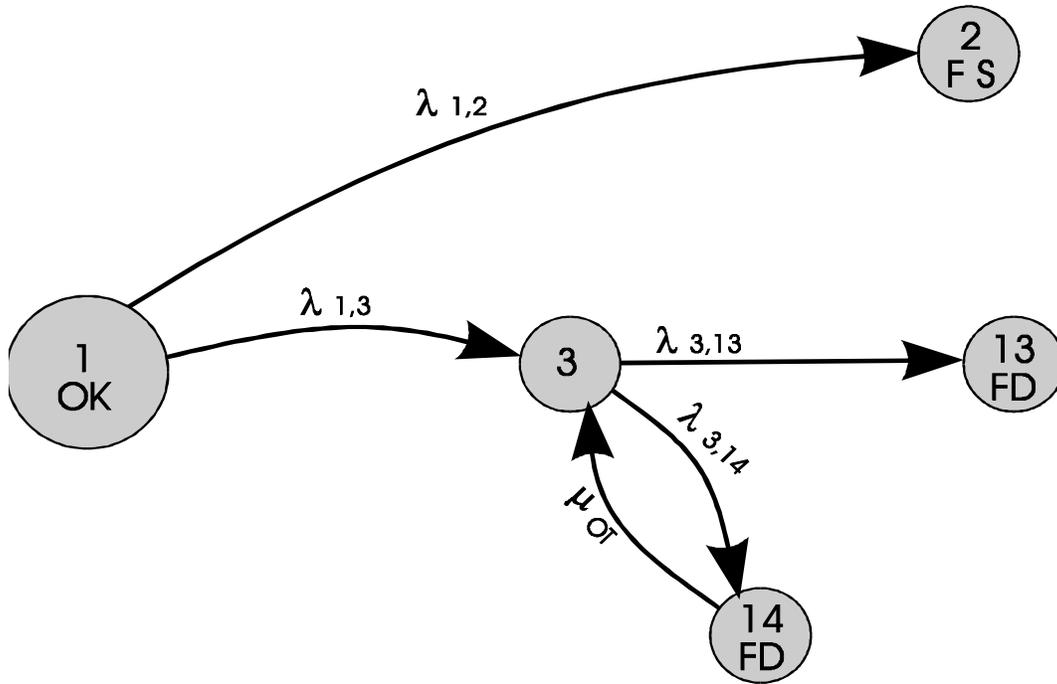


Figure A.9 — Markov model for transition from state 3 to state 13 elimination of transition from same state to the much smaller two failed states

The transition that dominates the failure of the logic solver to state 2 is λ_1 (see Equations 20 and 21). Using the same concept, the transition that dominates the failure of the logic solver from state 3 to state 13 is $\lambda_{3,13}$. The resulting Markov models after the above assumptions have been implemented is shown in Figure A.10.

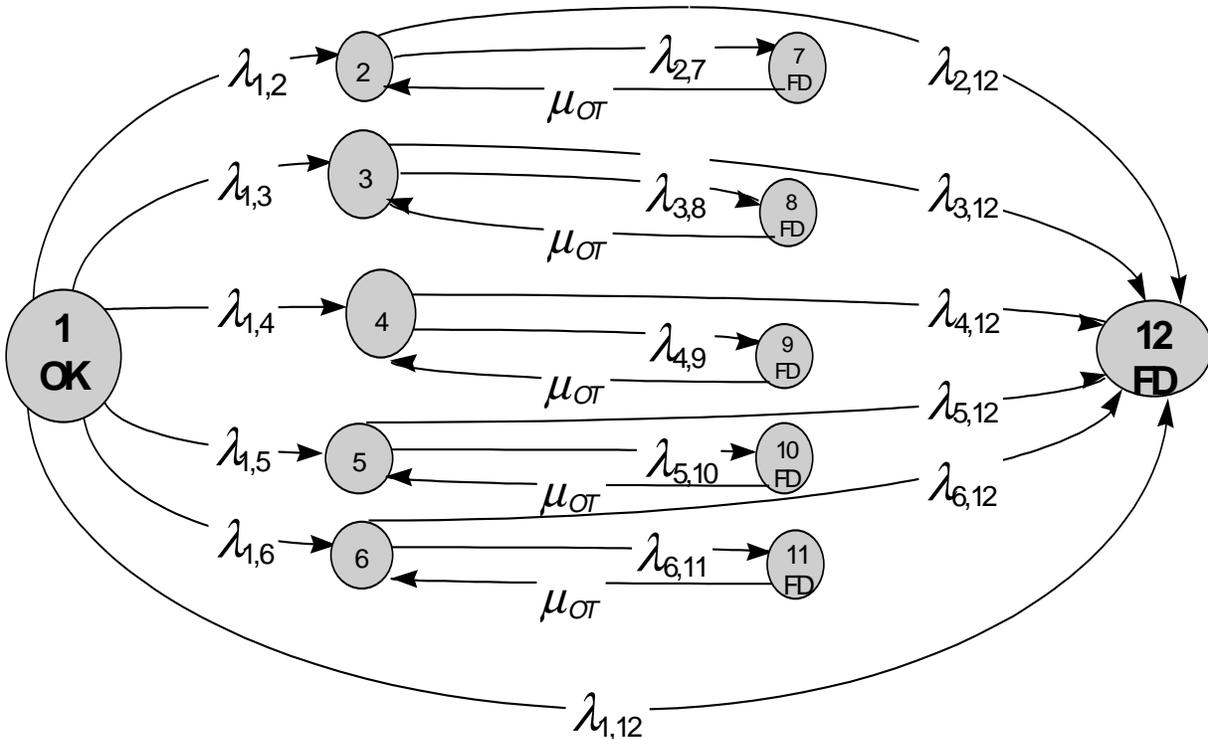


Figure A.10 — Markov model for transition to state 2 using dominant λ_1

A.3 Assumptions and limitations

The assumptions made during the Markov model development are listed in Clause 4.3. The impact on the models if these assumptions are changed is also discussed.

A.4 Quantification of a Markov model

The generalized **fail-dangerous** Markov model is shown in Figure A.10. Each state represents the PE logic solver under specific conditions (i.e., component failures). For example, state 1 represents the state of the logic solver where all components are functioning properly. A failure of a component will force the logic solver to transition (fail) to one of the states defined as 2-6 (inclusive). For example, a failure of an input microprocessor will send the logic solver from state 1 to state 2. Component failures in this sequence will eventually put the logic solver in state 12, which is the failed dangerous state for the PE logic solver.

For the general model

- state 1 represents a fully operational logic solver,

- states 2-6 represent degraded but still functional states of the PE logic solver, and
- states 7-12 represent failed states of the PE logic solver.

Therefore, the Markov model is a state space diagram that defines the functional conditions of the PE logic solver (state) in time (space). What this clause will provide is a methodology to evaluate the probability of a PE logic solver to be found in any of the 12 states within a specific time interval.

The parameters of interest are the rates or probabilities of transition from one state to another. The rates are λ and μ , where λ is the failure rate and μ is the repair rate of a component. The probability of transition (probability to have a failure or a repair) in a time interval from time $t=0$ to time t , is given as:

$$(Eq. A.36) \quad P(failure) = 1 - e^{-\lambda t}$$

$$(Eq. A.37) \quad P(repair) = 1 - e^{-\mu t}$$

Expanding the exponential term, Equation 36 may be written as

$$(Eq. A.38) \quad P(failure) = 1 - [1 - \lambda t + \sum_{n=2}^{\infty} \frac{(-\lambda t)^n}{n!}]$$

For $\lambda t < 0.1$, Equation 38 reduces to:

$$(Eq. A.39) \quad P(failure) = \lambda t$$

Equation 39 gives the rare event approximation and is a parameter that is used for the transition probabilities in the Markov model. Consider now a small interval of time Δt which is made sufficiently small so that the probability of two or more transitions occurring during Δt is negligible. The probability to have a failure is given by Equation 36. The probability to have a repair is similar to Equation 36. Simply replace λ with μ .

On Figure A.10, the transition probabilities are shown with arcs. Each arc is labeled by its transition probability. Due to convention, the time is not shown on the models.

Two methods can be used to solve this general model and determine the probability of the system to be found in any state in a given time interval. These methods are: The Differential Equations Method and The Matrix Multiplication Method (Numerical Solutions).

A.4.1 Differential equations method

A.4.1.1 Probability to fail on demand (PFD)

For the PE logic solver in Figure A.10, the probability of being in state 1 at time $t+\Delta t$ and remaining in this state at time $t+\Delta t$ (i.e., the probability that the system does not transition out of state 1 at time Δt , given that the system was in state 1 at time t) is:

Probability of being in state 1 at $t+\Delta t$ = [Probability of being in state 1 at time t AND of not failing in time Δt] + [Probability of being failed (being in another state that communicates with state 1 at time t AND of being repaired back to state 1 in time Δt]

From the model in Figure A.10, the probability of the PE logic solver to be in state 1 at time $t+\Delta t$ is:

$$(Eq. A.40) \quad P_1(t + \Delta t) = P_1(t)[1 - (\lambda_{1,2} + \lambda_{1,3} + \lambda_{1,4} + \lambda_{1,5} + \lambda_{1,6} + \lambda_{1,12})\Delta t]$$

Take P_1 to the other side of Equation 40,

$$(Eq. A.41) \quad P_1(t + \Delta t) - P_1(t) = -[\lambda_{1,2} + \lambda_{1,3} + \lambda_{1,4} + \lambda_{1,5} + \lambda_{1,6} + \lambda_{1,12}]\Delta t P_1(t)$$

Dividing both sides of Equation 41 by Δt and taking the limit as Δt approaches zero, Equation 41 can be written as:

$$(Eq. A.42) \quad \lim_{\Delta t \rightarrow 0} \frac{P_1(t + \Delta t) - P_1(t)}{\Delta t} = \lambda_{1,1} P_1(t)$$

where $\lambda_{1,1}$ is defined as

$$(Eq. A.43) \quad \lambda_{1,1} = -[\lambda_{1,2} + \lambda_{1,3} + \lambda_{1,4} + \lambda_{1,5} + \lambda_{1,6} + \lambda_{1,12}]$$

Therefore, Equation 42 can be written as

$$(Eq. A.44) \quad \dot{P}_1 = \frac{dP_1(t)}{dt} = \lambda_{1,1} P_1(t)$$

Similar relationships exist for the other states. For example, the transition probability rate for state 2 is determined as follows:

$$(Eq. A.45) \quad P_2(t + \Delta t) = P_2(t)[1 - (\lambda_{2,12} + \lambda_{2,7})\Delta t] + \lambda_{1,2} \Delta t P_1(t) + \mu_{OT} \Delta t P_7(t)$$

or,

$$(Eq. A.46) \quad \dot{P}_2 = \frac{dP_2(t)}{dt} = -(\lambda_{2,12} + \lambda_{2,7})P_2(t) + \lambda_{1,2} P_1(t) + \mu_{OT} P_7(t)$$

Similarly, the remaining state equations are:

$$(Eq. A.47) \quad \dot{P}_3 = \frac{dP_3(t)}{dt} = -(\lambda_{3,12} + \lambda_{3,8})P_3(t) + \lambda_{1,3} P_1(t) + \mu_{OT} P_8(t)$$

$$(Eq. A.48) \quad \dot{P}_4 = \frac{dP_4(t)}{dt} = -(\lambda_{4,12} + \lambda_{4,9})P_4(t) + \lambda_{1,4}P_1(t) + \mu_{OT}P_9(t)$$

$$(Eq. A.49) \quad \dot{P}_5 = \frac{dP_4(t)}{dt} = -(\lambda_{4,12} + \lambda_{4,9})P_4(t) + \lambda_{1,4}P_1(t) + \mu_{OT}P_9(t)$$

$$(Eq. A.1) \quad \dot{P}_6 = \frac{dP_6(t)}{dt} = -(\lambda_{6,12} + \lambda_{6,11})P_6(t) + \lambda_{1,6}P_1(t) + \mu_{OT}P_{11}(t)$$

$$(Eq. A.51) \quad \dot{P}_7 = \frac{dP_7(t)}{dt} = -\mu_{OT}P_7(t) + \lambda_{2,7}P_2(t)$$

$$(Eq. A.52) \quad \dot{P}_8 = \frac{dP_8(t)}{dt} = -\mu_{OT}P_8(t) + \lambda_{3,8}P_3(t)$$

$$(Eq. A.53) \quad \dot{P}_9 = \frac{dP_9(t)}{dt} = -\mu_{OT}P_9(t) + \lambda_{4,9}P_4(t)$$

$$(Eq. A.54) \quad \dot{P}_{10} = \frac{dP_{10}(t)}{dt} = -\mu_{OT}P_{10}(t) + \lambda_{5,10}P_5(t)$$

$$(Eq. A.55) \quad \dot{P}_{11} = \frac{dP_{11}(t)}{dt} = -\mu_{OT}P_{11}(t) + \lambda_{6,11}P_6(t)$$

$$(Eq. A.56) \quad \dot{P}_{12} = \lambda_{1,12}P_1(t) + \lambda_{2,12}P_2(t) + \lambda_{3,12}P_3(t) + \lambda_{4,12}P_4(t) + \lambda_{5,12}P_5(t) + \lambda_{6,12}P_6(t)$$

From the above analysis, the probability that the logic solver will fail on demand (PFD) is the probability that the PE logic solver will be found in any of the failed states during a time interval. Therefore, since the states 7, 8, 9, 10, 11 and 12 have been defined as failed states, PFD is given as:

$$(Eq. A.57) \quad PFD(t) = P_7(t) + P_8(t) + P_9(t) + P_{10}(t) + P_{11}(t) + P_{12}(t)$$

To determine the variables in Equation 57 requires the simultaneous solution of the 12 differential state equations. First solve the equation for state 1 which is a simple differential equation given by equation 44. There are several methods to solve this equation. One is Laplace transformation, which will be used to illustrate the solution for this simple equation. The same type of transformations will be used later for the more complex differential equations.

The Laplace transformation of any derivative is given below:

$$(Eq. A.58) \quad \dot{P}_1 = \frac{dP_1(t)}{dt} = sP_1(s) - P_1(0)$$

where t is replaced by s in the Laplace domain, and $P_1(0)$ is the initial condition evaluated at time $t=0$. Therefore, Equation 44 transformed into the Laplace domain is:

$$(Eq. A.59) \quad sP_1(s) - P_1(0) = \lambda_{1,1} P_1(s)$$

Using the initial condition that at $t=0$ the logic solver is in state 1, $P_1(0)=1$, Equation 59 reduces to:

$$(Eq. A.60) \quad sP_1(s) - 1 = \lambda_{1,1} P_1(s)$$

Solving Equation 60 for $P_1(s)$ we obtain:

$$(Eq. A.61) \quad P_1(s) = \frac{1}{(s - \lambda_{1,1})}$$

Having solved the linear equation in the Laplace domain, we must transform this equation back to the real time domain. This transformation can be accomplished using functions that can be found in any calculus book. The transformation for equations similar to Equation 61 is:

$$(Eq. A.62) \quad f(s) = \frac{1}{s - a} \quad \text{Then} \quad F(t) = e^{at}$$

Equation 61 transforms to:

$$(Eq. A.63) \quad P_1(t) = e^{\lambda_{1,1}t}$$

The exponent of Equation 63 is actually negative and is given by Equation 43.

There are several states, namely 7, 8, 9, 10 and 11, that communicate only with one other state. For example, state 7 communicates only with state 2. In addition, the probability that the logic solver is in state 7 at time $t+\Delta t$ is a function of the probability that the logic solver was in state 2 at time t and made the transition to state 7 at time $t+\Delta t$. Since the transition rate from state 2 to state 7, $\lambda_{(2,7)}$, is typically much smaller than the on-line repair rate, μ_{OT} , from state 7 back to state 2, it can be assumed that:

$$(Eq. A.64) \quad P_7(t) = \frac{\lambda_{2,7} P_2(t)}{\mu_{OT}}$$

$$(Eq. A.65) \quad P_8(t) = \frac{\lambda_{3,8} P_3(t)}{\mu_{OT}}$$

$$(Eq. A.66) \quad P_9(t) = \frac{\lambda_{4,9} P_4(t)}{\mu_{OT}}$$

$$(Eq. A.67) \quad P_{10}(t) = \frac{\lambda_{5,10} P_5(t)}{\mu_{OT}}$$

$$(Eq. A.68) \quad P_{11}(t) = \frac{\lambda_{6,11} P_6(t)}{\mu_{OT}}$$

This assumption will eliminate the effect of these states on state 12 as it will be shown later in this clause, but will slightly adjust the probability of the logic solver to fail on demand given by Equation 57.

Now the remaining state equations must be solved. Take the equation for state 2 (Equation 11) and replace the term $P_7(t)$ with Equation 64. Equation 46 is changed to:

$$(Eq. A.69) \quad \dot{P}_2 = -(\lambda_{2,12} + \lambda_{2,7}) P_2(t) + \lambda_{1,2} P_1(t) + \frac{\mu_{OT} \lambda_{2,7} P_2(t)}{\mu_{OT}}$$

Equation 69 reduces to the following form by eliminating μ_{OT} and $\lambda_{2,7} P_2(t)$.

Following the same approach for states 3, 4, 5 and 6, their state equations reduce to the following form:

$$(Eq. A.70) \quad \dot{P}_2 = -\lambda_{2,12} P_2(t) + \lambda_{1,2} P_1(t)$$

$$(Eq. A.71) \quad \dot{P}_3 = -\lambda_{3,12} P_3(t) + \lambda_{1,3} P_1(t)$$

$$\text{(Eq. A.72)} \quad \dot{P}_4 = -\lambda_{4,12} P_4(t) + \lambda_{1,4} P_1(t)$$

$$\text{(Eq. A.73)} \quad \dot{P}_5 = -\lambda_{5,12} P_5(t) + \lambda_{1,5} P_1(t)$$

$$\text{(Eq. A.74)} \quad \dot{P}_6 = -\lambda_{6,12} P_6(t) + \lambda_{1,6} P_1(t)$$

Each of these Equations (70-74) must be solved. The same procedure of Laplace transformation will be used. To solve the equation for state 2, replace in Equation 70 the term $P_1(t)$ with Equation 63. Hence, Equation 70 can be written as:

$$\text{(Eq. A.75)} \quad \dot{P}_2 = -\lambda_{2,12} P_2(t) + \lambda_{1,2} e^{\lambda_{1,1} t}$$

Transfer Equation 75 into the Laplace domain using the transform functions given by Equations 58 and 61.

$$\text{(Eq. A.76)} \quad sP_2(s) - P_2(0) = -\lambda_{2,12} P_2(s) + \frac{\lambda_{1,2}}{s - \lambda_{1,1}}$$

The initial condition for state 2 is $P_2(0)=0$. Use the initial condition in Equation 76 and solve for $P_2(s)$.

$$\text{(Eq. A.77)} \quad P_2(s) = \frac{\lambda_{1,2}}{(s - \lambda_{2,12})(s - \lambda_{1,1})}$$

Equation 77 must be transferred back to the time domain using the following transformation:

If the Laplace function has the following form,

$$\text{(Eq. A.78)} \quad f(s) = \frac{1}{(s-a)(s-b)}$$

then the time dependent function takes the form,

$$\text{(Eq. A.79)} \quad F(t) = \frac{1}{(a-b)} [e^{at} - e^{bt}]$$

Letting $a = -\lambda_{1,12}$ and $b = \lambda_{1,1}$, the time dependent function of Equation 77 is given as:

$$(Eq. A.80) \quad P_2(t) = \frac{-\lambda_{1,2}}{(\lambda_{1,1} + \lambda_{2,12})} [e^{-\lambda_{1,12}t} - e^{\lambda_{1,1}t}]$$

Similarly, Equations 71-74 can be solved and the time dependent functions are given below:

$$(Eq. A.81) \quad P_3(t) = \frac{-\lambda_{1,3}}{(\lambda_{1,1} + \lambda_{3,12})} [e^{-\lambda_{3,12}t} - e^{\lambda_{1,1}t}]$$

$$(Eq. A.82) \quad P_4(t) = \frac{-\lambda_{1,4}}{(\lambda_{1,1} + \lambda_{4,12})} [e^{-\lambda_{4,12}t} - e^{\lambda_{1,1}t}]$$

$$(Eq. A.83) \quad P_5(t) = \frac{-\lambda_{1,5}}{(\lambda_{1,1} + \lambda_{5,12})} [e^{-\lambda_{5,12}t} - e^{\lambda_{1,1}t}]$$

$$(Eq. A.84) \quad P_6(t) = \frac{-\lambda_{1,6}}{(\lambda_{1,1} + \lambda_{6,12})} [e^{-\lambda_{6,12}t} - e^{\lambda_{1,1}t}]$$

These are the probability functions for states 1-6. State 12 is given by Equation 56. Since all the state equations have been solved (i.e., $P_1(t)$, $P_2(t)$, ..., $P_6(t)$ are known), simply replace these variables in Equation 56 by their respective function given in Equations 80-84 and integrate the function from $t=0$ to $t=t$.

$$(Eq. A.85) \quad P_{12}(t) = \int_0^t [\lambda_{1,12} P_1(t) + \lambda_{2,12} P_2(t) + \lambda_{3,12} P_3(t) + \lambda_{4,12} P_4(t) + \lambda_{5,12} P_5(t) + \lambda_{6,12} P_6(t)] dt$$

Each parameter in Equation 85 can be integrated separately. The solution to Equation 85 is given below:

$$(Eq. A.86) \quad P_{12}(t) = A(t) + B(t) + C(t) + D(t) + E(t) + F(t)$$

$$(Eq. A.87) \quad A(t) = \frac{-\lambda_{1,12}}{\lambda_{1,1}} [1 - e^{\lambda_{1,1}t}]$$

$$(Eq. A.88) \quad B(t) = \frac{-\lambda_{1,2} \lambda_{2,12}}{\lambda_{1,1} + \lambda_{2,12}} \left[\frac{1 - e^{-\lambda_{2,12}t}}{\lambda_{2,12}} + \frac{1 - e^{-\lambda_{1,1}t}}{\lambda_{1,1}} \right]$$

$$(Eq. A.89) \quad C(t) = \frac{-\lambda_{1,3} \lambda_{3,12}}{\lambda_{1,1} + \lambda_{3,12}} \left[\frac{1 - e^{-\lambda_{3,12}t}}{\lambda_{3,12}} + \frac{1 - e^{-\lambda_{1,1}t}}{\lambda_{1,1}} \right]$$

$$(Eq. A.90) \quad D(t) = \frac{-\lambda_{1,4} \lambda_{4,12}}{\lambda_{1,1} + \lambda_{4,12}} \left[\frac{1 - e^{-\lambda_{4,12}t}}{\lambda_{4,12}} + \frac{1 - e^{-\lambda_{1,1}t}}{\lambda_{1,1}} \right]$$

$$(Eq. A.91) \quad E(t) = \frac{-\lambda_{1,5} \lambda_{5,12}}{\lambda_{1,1} + \lambda_{5,12}} \left[\frac{1 - e^{-\lambda_{5,12}t}}{\lambda_{5,12}} + \frac{1 - e^{-\lambda_{1,1}t}}{\lambda_{1,1}} \right]$$

$$(Eq. A.92) \quad F(t) = \frac{-\lambda_{1,6} \lambda_{6,12}}{\lambda_{1,1} + \lambda_{6,12}} \left[\frac{1 - e^{-\lambda_{6,12}t}}{\lambda_{6,12}} + \frac{1 - e^{-\lambda_{1,1}t}}{\lambda_{1,1}} \right]$$

The remaining state functions, $P_7(t)$ to $P_{11}(t)$ given by Equations 64-68, must now be determined. Take the equation for state 7, Equation 64, and replace $P_2(t)$ with Equation 80. The equation for state 7 transforms to the following form:

$$(Eq. A.93) \quad P_7(t) = \left[\frac{\lambda_{2,7}}{\mu_{OT}} \right] \left[-\frac{\lambda_{1,2}}{\lambda_{1,1} + \lambda_{2,12}} (e^{-\lambda_{2,12}t} - e^{-\lambda_{1,1}t}) \right]$$

Rearranging terms in Equation 93, we obtain the final form for $P_7(t)$, which is given below:

$$(Eq. A.94) \quad P_7(t) = \frac{-\lambda_{2,7} MTTR_{OT} \lambda_{1,2}}{\lambda_{1,1} + \lambda_{2,12}} (e^{-\lambda_{2,12}t} - e^{-\lambda_{1,1}t})$$

Similar formulations for $P_8(t)$ to $P_{11}(t)$ give the following:

$$(Eq. A.95) \quad P_8(t) = \frac{-\lambda_{3,8} MTTR_{OT} \lambda_{1,3}}{\lambda_{1,1} + \lambda_{3,12}} (e^{-\lambda_{3,12}t} - e^{-\lambda_{1,1}t})$$

$$(Eq. A.96) \quad P_9(t) = \frac{-\lambda_{4,9} MTTR_{OT} \lambda_{1,4}}{\lambda_{1,1} + \lambda_{4,12}} (e^{-\lambda_{4,12}t} - e^{-\lambda_{1,1}t})$$

$$(Eq. A.97) \quad P_{10}(t) = \frac{-\lambda_{5,10} MTTR_{OT} \lambda_{1,5}}{\lambda_{1,1} + \lambda_{5,12}} (e^{-\lambda_{5,12}t} - e^{-\lambda_{1,1}t})$$

$$(Eq. A.98) \quad P_{11}(t) = \frac{-\lambda_{6,11} MTTR_{OT} \lambda_{1,6}}{\lambda_{1,1} + \lambda_{6,12}} (e^{-\lambda_{6,12}t} - e^{-\lambda_{1,1}t})$$

The probability that the PE logic solver is at any state as a function of time can now be determined from Equations 62, 80-84, 86, and 94-98. The probability that the PE logic solver will fail on demand, PFD is still given by Equation 57.

There are two limiting conditions that the solutions must satisfy. The conditions are that at $t=0$ $P_1(0)=1$ and at $t=\infty$ then $P_{12}=1$.

First investigate the limiting condition at $t=0$. By inspecting Equations 62, 80-84, 86 and 94-98, we see that all the terms are eliminated except for Equation 62 which becomes,

$$(Eq. A.99) \quad P_1(0) = e^{\lambda_{1,0}} = 1$$

In order to check the limiting condition that time approaches infinity, the terms in Equation 86 must be rearranged to give a function that has a constant term followed by six terms that are exponential in form and all with negative exponents. Letting time approach infinity, all the exponential terms are eliminated from the equation. That leaves only the constant term given below:

$$(Eq. A.100) \quad P_{12}(\infty) = -\frac{I}{\lambda_{1,1}} [\lambda_{1,2} + \lambda_{1,3} + \lambda_{1,4} + \lambda_{1,5} + \lambda_{1,6} + \lambda_{1,12}]$$

The term in the brackets is defined by Equation 43. Therefore, Equation 100 reduces to:

$$(Eq. A.101) \quad P_{12}(\infty) = 1$$

A.4.1.2 Mean time to failure dangerous (MTTF_D)

The definition of MTTF_D is given as,

$$(Eq. A.102) \quad MTTF_D = \int_0^{\infty} R(t)dt$$

where, $R(t)$ is the reliability of the logic solver. For the generalized model, the reliability is the probability that the PE logic solver is found in any of the working states (1-6 inclusive) and is given by the following equation:

$$(Eq. A.103) \quad R(t) = P_1(t) + P_2(t) + P_3(t) + P_4(t) + P_5(t) + P_6(t)$$

where $P_1(t)$ to $P_6(t)$ is given by Equations 62 and 80-84. Equation 102 can be written as:

$$(Eq. A.104) \quad MTTF_D = \int_0^{\infty} [P_1(t) + P_2(t) + P_3(t) + P_4(t) + P_5(t) + P_6(t)] dt$$

Integrating Equation 104, the $MTTF_D$ for the general Markov model is obtained:

$$(Eq. A.105) \quad MTTF_D = -\frac{1}{\lambda_{1,1}} \left[1 + \frac{\lambda_{1,2}}{\lambda_{2,12}} + \frac{\lambda_{1,3}}{\lambda_{3,12}} + \frac{\lambda_{1,4}}{\lambda_{4,12}} + \frac{\lambda_{1,5}}{\lambda_{5,12}} + \frac{\lambda_{1,6}}{\lambda_{6,12}} \right]$$

It can be observed that the $MTTF_D$ is dominated by $\lambda_{1,1}$ in Equation 105.

A.4.2 Matrix multiplication method

A.4.2.1 Probability to fail on demand (PFD)

For many complex systems, the differential equation method described in A.4.1 becomes very complex and cumbersome. In such cases, numerical solutions are available using the Matrix Multiplication Method.

In order to apply this method to the PE logic solver in Figure A.10, it is necessary to deduce a matrix, which represents the probabilities of making a transition from one state to another in a single time interval (time step) Δt . This time interval must be sufficiently small such that the probability of making two or more transitions (component failures or repairs) in this time interval is negligible.

Define t_{ij} as the probability of making a transition to state j after a time interval Δt given that the PE logic solver was in state i at the beginning of the time interval.

Therefore, t_{ij} is a dependent probability that can be defined as:

$$(Eq. A.106) \quad t_{ij}(t + \Delta t) = \text{Probability}(\text{state } j \text{ at } t + \Delta t \mid \text{state } i \text{ at } t)$$

The above definition also indicates that the row position of the matrix to be developed is the state from which the transition occurs and the column position of the matrix is the state to which the transition leads the PE logic solver. Consequently for any n -state system (Markov model), the general form of the matrix must also be square since the system can transition from any state to any other state. In addition, transitions that are impossible to be made have a transition rate equal to zero. The general form of the matrix is given below:

MATRIX T		TO STATE				
		1	2	3	n
FROM STATE	1	t _{1,1}	t _{1,2}	t _{1,3}	t _{1,n}
	2	t _{2,1}	t _{2,2}	t _{2,3}	t _{2,n}
	3	t _{3,1}	t _{3,2}	t _{3,3}	t _{3,n}

	n	t _{n,1}	t _{n,2}	t _{n,3}	t _{n,n}

The above **T** matrix is a stochastic transition probability matrix. The **T** matrix can be further separated into the dangerous and safe transition matrix, **T^D** and **T^S**, respectively. It should be noted that the summation of the probabilities in each row of the matrix must be unity since row *i* represents the complete and exhaustive ways in which the system can behave in a particular time interval given that it was in state *i* at the beginning of the interval. Therefore,

$$(Eq. A.107) \quad \sum_{j=1}^{j=n} t_{ij} = 1$$

The conditional probabilities, *t_{ij}*, can be determined from a state balance equation. The method to determine the balance equations is illustrated with an example. Suppose the balance equation for state 2 is needed. The process enters state 2 either from state 1 or state 7. Thus the probability of being in state 1, *P₁*, represents the portion of the time that it would be possible for the process to enter state 2 from state 1. Similarly, *P₇* is the portion of the time that the process would enter state 2 from state 7. Given that the process is in state 1, the rate of entering state 2 is $\lambda_{1,2}$ and the probability of transition from state 1 to state 2 is,

$$(Eq. A.108) \quad t_{1,2} = \lambda_{1,2} \Delta t$$

$$(Eq. A.109) \quad t_{7,2} = \lambda_{7,2} \Delta t$$

Similar arrangements exist for the probability to transition from state 7 to state 2, given that the logic solver was in state 7.

The logic solver can transition out of state 2 (to state 12 or 7) with leaving transition rates $\lambda_{2,7}$ and $\lambda_{2,12}$. Therefore, the probability to transition from state 2 to state 12 in time Δt given that the logic solver is in state 2 at time *t*, is:

$$(Eq. A.110) \quad t_{2,12} = \lambda_{2,12} \Delta t$$

The logic solver can remain in state 2 after a time interval Δt given that the logic solver was in state 2 at the beginning of the interval. This is simply stating that there is a probability that the logic solver being in state 2 will not fail to either state 7 or 12 and is given as:

$$(Eq. A.111) \quad P_{2,2} = I - [\lambda_{2,12} + \lambda_{2,7}] \Delta t$$

The first term in the right hand side of Equation 111 is the probability that the logic solver was in state 2 at time t , the second term is the probability that the logic solver failed to state 12 in time Δt and the third term is the probability that the logic solver failed to state 7 in time Δt .

$$(Eq. A.112) \quad t_{1,1} = I - [\lambda_{1,2} + \lambda_{1,3} + \lambda_{1,4} + \lambda_{1,5} + \lambda_{1,6} + \lambda_{1,12}] \Delta t = I - \lambda_{1,1} \Delta t$$

$$(Eq. A.113) \quad t_{2,2} = I - [\lambda_{2,12} + \lambda_{2,7}] \Delta t$$

$$(Eq. A.114) \quad t_{3,3} = I - [\lambda_{3,12} + \lambda_{3,8}] \Delta t$$

$$(Eq. A.115) \quad t_{4,4} = I - [\lambda_{4,12} + \lambda_{4,9}] \Delta t$$

$$(Eq. A.116) \quad t_{5,5} = I - [\lambda_{5,12} + \lambda_{5,10}] \Delta t$$

$$(Eq. A.117) \quad t_{6,6} = I - [\lambda_{6,12} + \lambda_{6,11}] \Delta t$$

$$(Eq. A.118) \quad t_{7,7} = I - \lambda_{7,2} \Delta t = I - \mu_{OT} \Delta t$$

$$(Eq. A.119) \quad t_{8,8} = I - \lambda_{8,3} \Delta t = I - \mu_{OT} \Delta t$$

$$(Eq. A.120) \quad t_{9,9} = I - \lambda_{9,4} \Delta t = I - \mu_{OT} \Delta t$$

$$(Eq. A.121) \quad t_{10,10} = I - \lambda_{10,5} \Delta t = I - \mu_{OT} \Delta t$$

$$(Eq. A.122) \quad t_{11,11} = I - \lambda_{11,6} \Delta t = I - \mu_{OT} \Delta t$$

$$(Eq. A.123) \quad t_{12,12} = I$$

For the model in Figure A.10, the conditional probabilities from matrix **T** are:

Once the logic solver enters state 12 it does not leave the state, therefore state 12 is an absorbing state.

$$(Eq. A.124) \quad t_{2,7} = \lambda_{2,7} \Delta t$$

$$(Eq. A.125) \quad t_{3,8} = \lambda_{3,8} \Delta t$$

$$(Eq. A.126) \quad t_{4,9} = \lambda_{4,9} \Delta t$$

$$(Eq. A.127) \quad t_{5,10} = \lambda_{5,10} \Delta t$$

$$(Eq. A.128) \quad t_{6,11} = \lambda_{6,11} \Delta t$$

$$(Eq. A.129) \quad t_{7,2} = t_{8,3} = t_{9,4} = t_{10,5} = t_{11,6} = \mu_{OT} \Delta t$$

$$(Eq. A.130) \quad t_{2,12} = \lambda_{2,12} \Delta t$$

$$(Eq. A.131) \quad t_{3,12} = \lambda_{3,12} \Delta t$$

$$(Eq. A.132) \quad t_{4,12} = \lambda_{4,12} \Delta t$$

$$(Eq. A.133) \quad t_{5,12} = \lambda_{5,12} \Delta t$$

$$(Eq. A.134) \quad t_{6,12} = \lambda_{6,12} \Delta t$$

Equations 112-134 are the cells of the **T** matrix that are greater than zero. All other cells are zero because no transition exists for those states. As was stated earlier, in order for this matrix to be a stochastic probability matrix, the sum of the cells in each row must add to unity. For example, the sum of all cells in row 3 is given as,

$$\text{(Eq. A.135)} \quad t_{3,1} + t_{3,2} + \dots + t_{3,12} = I - (\lambda_{3,12} + \lambda_{3,8})\Delta t + \lambda_{3,8} \Delta t + \lambda_{3,12} \Delta t = I$$

It is important to note that Δt must be included in the formulation of Equations 112-134. If a value of $\Delta t = 1$ (time units) is used, then Δt is usually omitted from the equations for simplicity.

Having defined the stochastic transition matrix, **T**, the probability of the system after k time intervals can be determined using the following relationship:

$$\text{(Eq. A.136)} \quad T^{(k)} = T^k$$

Equation 136 simply states that the system state probabilities after k time intervals, $P^{(k)}$, can be determined by taking the original **T** matrix to the k^{th} power. Make certain that the units of k are consistent with that of Δt . For example, if $\Delta t = 1$ hour, and the system is to be evaluated for 1 year, then $k = 8640$ hours and the matrix must be raised to the 8640th power.

$$\text{(Eq. A.137)} \quad T^{(k)} = T^{k-1} T$$

Equation 136 may be simplified and changed into a form more suitable for computer numerical analysis. This can be done to greatly reduce the CPU time requirements.

Equation 137 reduces the CPU time because the computer stores the previous result, P^{k-1} , and simply multiplies it with the original matrix.

The result from computations using either Equation 136 or 137 give a new stochastic probability matrix for the logic solver after k time intervals. This new matrix is of the same size as the original **T** matrix.

As was stated earlier, the results from the above analysis evaluate the conditional probabilities of the system to be in any state. In order to determine the unconditional probabilities, we must define the initial condition of the system. This is the condition of the PE logic solver at time $t=0$, and is typically assumed that the system is in state 1. Therefore, the initial condition of the system is given by a row matrix **Q** as shown below:

$$\text{(Eq. A.138)} \quad Q^{(0)} = [1 \ 0 \ \dots \ 0]$$

Equation 138 states that at time $t=0$, $P_1=1$ and all other $P_i=0$.

To determine the unconditional state probabilities for the PE logic solver the matrix given by Equation 137 is modified to read:

$$\text{(Eq. A.139)} \quad T^{(k)} = Q^{(0)} T^{(k-1)} T$$

Application of Equation 139 will result in a row vector whose cells give the probability for the PE logic solver to be in state *l* after *k* time intervals given that the system was in state 1 at time *t*=0.

(Eq. A.140) $P^{(k)} = [P_1 \ P_2 \ P_3 \ \dots\dots P_{12}]$

A.4.2.2 Mean time to failure dangerous (MTTF_D)

The mean time to failure for the PE logic solver can also be evaluated using the above formulations. Define a new matrix, **M**, as:

(Eq. A.141) $M = [I - R]^{-1}$

where **I** is the identity matrix and **R** is a new matrix obtained by deleting all the rows and columns from matrix **T** (Equation 107) associated with any absorbing states. In this formulation, the only absorbing state is state 12. Hence, row 12 and column 12 are eliminated from the **T** matrix to give us the new 11x11 matrix, **R**. The identity matrix, **I**, is also an 11x11 matrix.

Execution of Equation 141 will result in a new 11x11 matrix, **M**, similar to the one given below:

M =	<i>M</i> _{1,1}	<i>M</i> _{1,2}	<i>M</i> _{1,3}	<i>M</i> _{1,11}
	<i>M</i> _{2,1}	<i>M</i> _{2,2}	<i>M</i> _{2,3}	<i>M</i> _{2,11}
	<i>M</i> _{3,1}	<i>M</i> _{3,2}	<i>M</i> _{3,3}	<i>M</i> _{3,11}

	<i>M</i> _{11,1}	<i>M</i> _{11,2}	<i>M</i> _{11,3}	<i>M</i> _{11,11}

Each element in the **M** Matrix, for example, *M*_{3,1} is the average time the logic solver spent in state 1 given that the process starts at state 3 before being absorbed. Therefore, the sum of each row is the total average time of the PE logic solver before entering the absorbing state 12 given that the logic solver started in the state corresponding to that row. In the previous example, it was assumed that the PE logic solver was in state 1 at time *t*=0. Therefore, the MTTF of the PE logic solver can be determined as follows:

(Eq. A.142) $MTTF = \sum_{j=1}^{j=11} [M_{1,j}]$

If the initial conditions of the logic solver are used in the evaluation of logic solver MTTF,

(Eq. A.143) $MTTF = Q^{(0)} [I - R]^{-1}$

where $\mathbf{Q}^{<0>}$ is given by Equation 138, or the logic solver can be assumed to start from any state provided the sum of all the cells in the row matrix \mathbf{Q} add to unity. If we assume that the PE logic solver was in state 1 at time $t=0$, then Equations 107 and 108 will give the same results.

A.4.3 Mean time to failure for simple models

Some of the models in this document have only two states. State 1 corresponds to the logic solver being fully functional and state 2 corresponds to the logic solver having failed safe. Therefore, only one transition is possible, that from state 1 to state 2, given as $\lambda_{1,2}$. The mean time to failure for such a logic solver can be determined from Equation 102 and the probability of success of the logic solver given as:

$$(Eq. A.144) \quad R(t) = P_1(t) = e^{-\lambda_{1,2}t}$$

The MTTF is given as:

$$(Eq. A.145) \quad MTTF = \int_0^{\infty} R(t)dt = \int_0^{\infty} e^{-\lambda_{1,2}t} dt$$

Performing the integration in Equation 145 gives the well-known result for the MTTF of a simple logic solver.

$$(Eq. A.146) \quad MTTF = \frac{1}{\lambda_{1,2}}$$

A.4.4 Mean time to fail spurious for complex models

A typical Fail Spurious Markov model for the PE logic solver in this document is shown in Figure A.11. It should be noted that the Markov states $1_D, 1_U, 2_D, 2_U, \dots, N_D, N_U$ are for configurations where there is redundancy and two failures are required for the logic solver to be in the fail-safe state (state 0). It should also be noted that states $1_D, 2_D, \dots, N_D$ are for detected safe failures that are repaired on-line with repair rate μ_{OT} . States $1_U, 2_U, \dots, N_U$ are for undetected safe failures that are repaired during periodic off-line. In this document the assumption has been made that the mission time is the time between periodic off-line testing. Hence, the safe undetected failures that lead to states $1_U, 2_U, \dots, N_U$ cannot be repaired. The simplification of the Markov model is made in order to transform the model into Figure A.12 and which is the same form for which there exists a close form solution given below:

$$(Eq. A.147) \quad MTTF = \frac{1 + \sum_{i=1}^n \frac{\lambda_i}{\mu_i + \theta_i}}{\sum_{i=1}^n \frac{\lambda_i \theta_i}{\mu_i + \theta_i}}$$

where λ_i is the first component failure to an intermediate (degraded) state and θ_i is the failure rate from the degraded state i to the fail spurious state.

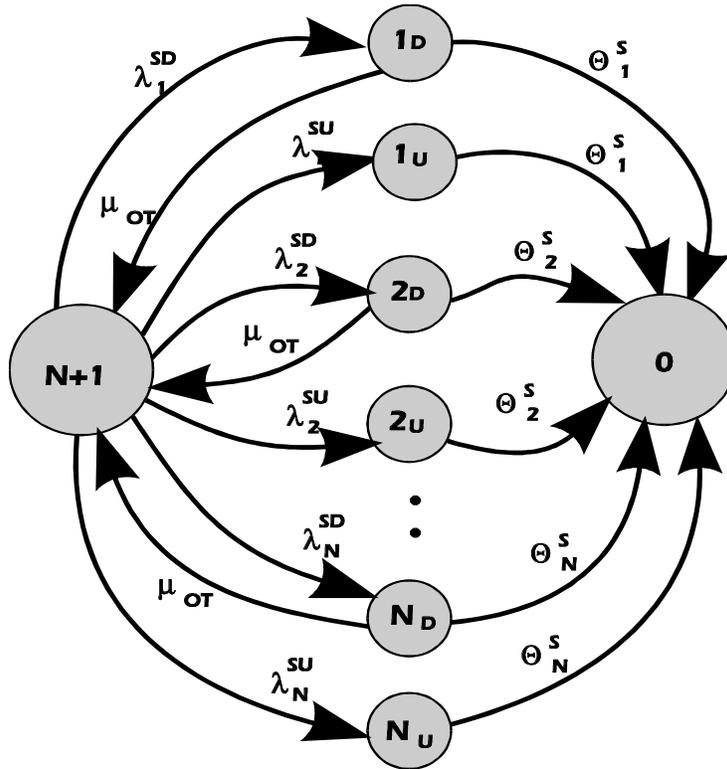


Figure A.11 — Markov model for PE logic solver with on-line diagnostics

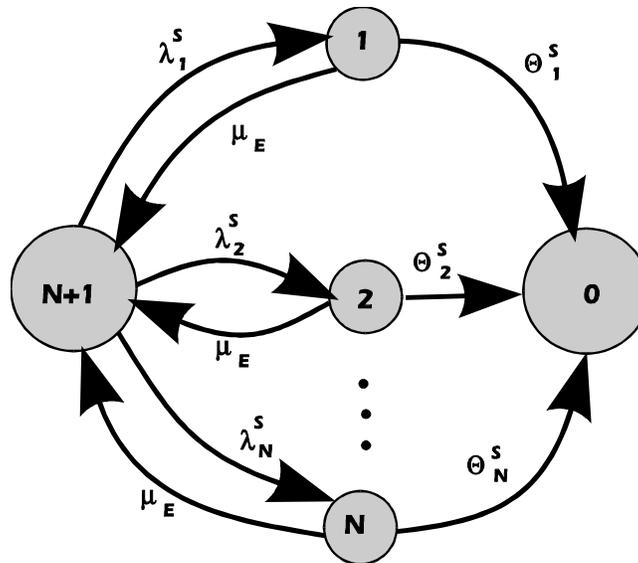


Figure A.12 — Markov model for PE logic solver with on-line diagnostics showing safe undetected failures cannot be repaired

Therefore, the simple Markov model shown in Figure A.13 must be transformed to the simple model shown in Figure A.14 by merging states 1_D and 1_U into one state, namely state 1. This is done by replacing the repair rate from the detected fail spurious state, μ_{OT} and the repair rate from the undetected fail spurious state which is zero, with an effective repair rate, μ_E . The remainder of the clause describes the method to determine the effective repair rate and thus merge the two states. This is accomplished using the differential equation method and by equating the mean time to failure, MTTF, of the two models.

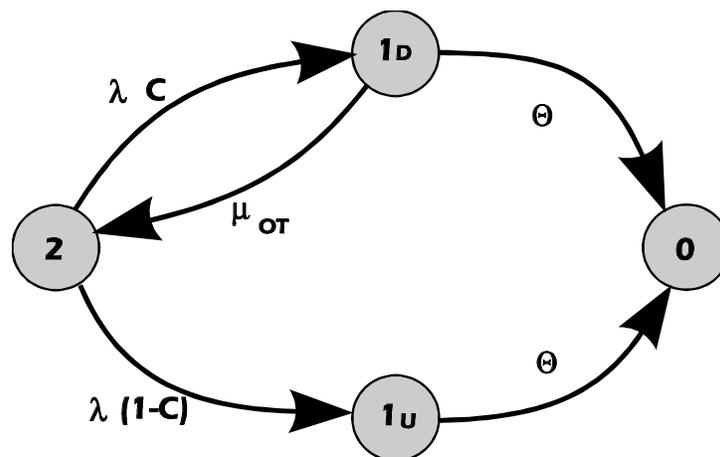


Figure A.13 — Markov model of PE logic solver with on-line diagnostics reduced to simple form

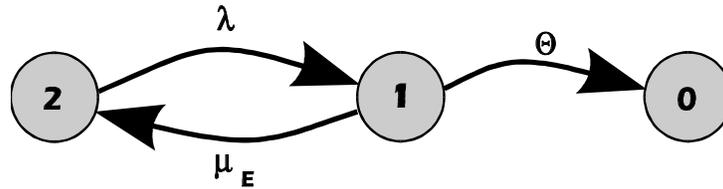


Figure A.14 — Simple form Markov model of PE logic solver with on-line diagnostics with a single repair rate μ_E

Following the same procedure as in Clause A.4 and working with the model in Figure A.13, the differential equations are:

$$(Eq. A.148) \quad \dot{P}_2 = -[\lambda C + \lambda(1 - C)] P_2(t) + \mu_{OT} P_{ID}(t)$$

$$(Eq. A.149) \quad \dot{P}_{ID} = -[\mu_{OT} + \theta] P_{ID}(t) + \lambda C P_2(t)$$

$$(Eq. A.150) \quad \dot{P}_{IU} = -\theta P_{IU}(t) + \lambda(1 - C) P_2(t)$$

$$(Eq. A.151) \quad \dot{P}_0 = \theta [P_{ID}(t) + P_{IU}(t)]$$

Transforming Equations 148-151 into the Laplace domain and using the initial condition that at time $t=0$ the PE logic solver is in state 2, we have,

$$(Eq. A.152) \quad sP_2(s) - 1 = -[\lambda C + \lambda(1 - C)] P_2(s) + \mu_{OT} P_{ID}(s)$$

$$(Eq. A.153) \quad sP_{IU}(s) = -\theta P_{IU}(s) + \lambda(1 - C) P_2(s)$$

$$(Eq. A.154) \quad sP_{ID}(s) = -[\mu_{OT} + \theta] P_{ID}(s) + \lambda C P_2(s)$$

$$(Eq. A.155) \quad sP_0(s) = \theta [P_{1D}(s) + P_{1U}(s)]$$

Solving Equations 153 and 154 for $P_{1D}(s)$ and $P_{1U}(s)$, respectively:

$$(Eq. A.156) \quad P_{1D}(s) = \frac{P_2(s)\lambda C}{(s + \mu_{OT} + \theta)}$$

$$(Eq. A.157) \quad P_{1U}(s) = \frac{P_2(s)\lambda(1-C)}{(s + \theta)}$$

Substitute Equations 156 and 157 into Equation 152 and solve for $P_2(s)$:

$$(Eq. A.158) \quad P_2(s) = \frac{(s + \mu_{OT} + \theta)}{[s + \lambda C + \lambda(1-C)](s + \mu_{OT} + \theta) - \lambda \mu_{OT} C}$$

The denominator in Equation 158 is a quadratic function of s with roots that can be easily obtained. For this analysis, the roots are not required. What will be required is the product of the roots. Therefore, Equation 158 may be written as:

$$(Eq. A.159) \quad P_2(s) = \frac{[s + \mu_{OT} + \theta]}{[s - D_1][s - D_2]}$$

Having determined $P_2(s)$, the remaining state equations may be solved by replacing $P_2(s)$ from Equation 159 into Equations 156, 157 and 154.

$$(Eq. A.160) \quad P_{1D}(s) = \frac{\lambda C}{[s - D_1][s - D_2]}$$

$$(Eq. A.161) \quad P_{1U}(s) = \frac{\lambda(1-C)(s + \mu_{OT} + \theta)}{[s - D_1][s - D_2][s + \theta]}$$

$$(Eq. A.162) \quad P_0(s) = \frac{\theta [\lambda C(s + \theta) + \lambda(1-C)(s + \mu_{OT} + \theta)]}{s[s - D_1][s - D_2][s + \theta]}$$

At this point the roots, (s-D₁) and (s-D₂) must be determined. Rewriting Equation 162 in partial fractions form and letting λ₁=λC and λ₂=λ(1-C), we have:

Solving Equation 163, we have:

$$(Eq. A.163) \quad \frac{\theta [\lambda_1(s + \theta) + \lambda_2(s + \mu_{OT} + \theta)]}{s[s - D_1][s - D_2][s + \theta]} = \frac{A}{s} + \frac{B}{s - D_1} + \frac{C}{s - D_2} + \frac{D}{s + \theta}$$

$$(Eq. A.164) \quad \begin{aligned} & \theta[\lambda_1(s + \theta) + \lambda_2(s + \mu_{OT} + \theta)] \\ & = A(s - D_1)(s - D_2)(s + \theta) + Bs(s - D_2)(s + \theta) + Cs(s - D_1)(s + \theta) + Ds(s - D_1)(s - D_2) \end{aligned}$$

Let s=0, Equation 164 reduces to:

$$(Eq. A.165) \quad \theta^2 \lambda_1 + \theta \lambda_2 (\mu_{OT} + \theta) = AD_1 D_2 \theta$$

The product of D₁D₂ must be determined. The logic solver has one absorbing state, state 0. Once the logic solver enters this state it cannot transition out. Therefore, as time approaches infinity (t→∞), the probability of the logic solver to reside in state 0 is unity, P₀(∞)=1. One way to determine D₁D₂θ is to determine the constant A in Equation 165 and solve 165 for the product D₁D₂θ. To determine A, take Equation 163 and transfer it back to the time domain using the transformation defined in Clause A.2. The probability function of state 0 is:

$$(Eq. A.166) \quad P_0(t) = A + Be^{D_1 t} + Ce^{D_2 t} + De^{-\theta t}$$

As time approaches infinity, Equation 166 reduces to:

$$(Eq. A.167) \quad P_0(\infty) = 1 = A$$

Having determined the value of A as time approaches infinity, and knowing that as time approaches infinity, the Laplace variable, s, approaches zero, Equation 165 may be solved for the product D₁D₂θ.

$$(Eq. A.168) \quad D_1 D_2 \theta = \theta^2 \lambda_1 + \lambda_2 \theta (\mu_{OT} + \theta)$$

Now the MTTF of the logic solver can be determined using the relationship given below:

$$(Eq. A.169) \quad MTTF_s = \int_0^{\infty} R(t)dt = \lim_{s \rightarrow 0} R(s)$$

The reliability of the logic solver is the probability to have the logic solver in any of the working states.

$$(Eq. A.170) \quad R(s) = P_2(s) + P_{1D}(s) + P_{1U}(s)$$

Replace the right hand side of Equation 170 with Equations 156, 157 and 158, and apply the limit shown in Equation 169 ($s=0$), then the $MTTF_s$ is:

$$(Eq. A.171) \quad MTTF_s = \frac{(\mu_{OT} + \theta)}{D_1 D_2} + \frac{\lambda_1}{D_1 D_2} + \frac{\lambda_2 (\mu_{OT} + \theta)}{D_1 D_2 \theta}$$

Rewrite Equation 171 so that the denominator is the product $D_1 D_2 \theta$ that is given in Equation 168,

$$(Eq. A.172) \quad MTTF_s = \frac{\theta (\mu_{OT} + \theta) + \theta \lambda_1 + \lambda_2 (\mu_{OT} + \theta)}{D_1 D_2 \theta}$$

Replace the denominator in Equation 172 by Equation 168, the $MTTF_s$ is:

$$(Eq. A.173) \quad MTTF_s = \frac{\theta (\mu_{OT} + \theta) + \theta \lambda_1 + \lambda_2 (\mu_{OT} + \theta)}{\theta [\lambda_1 \theta + \lambda_2 (\mu_{OT} + \theta)]}$$

Simplifying Equation 173, the mean time to failure is:

$$(Eq. A.174) \quad MTTF_s = \frac{(\mu_{OT} + \theta) + \lambda_1 + \frac{\lambda_2}{\theta} (\mu_{OT} + \theta)}{[\lambda_1 \theta + \lambda_2 (\mu_{OT} + \theta)]}$$

The $MTTF_s$ for the logic solver shown in Figure A.13 is given by Equation 174. As was stated earlier in this clause, the purpose of this analysis is to determine an effective repair rate, μ_E , in order to merge the states 1_D and 1_U . To do this, the logic solver in Figure A.14 must also be examined, its $MTTF_s$ determined and set equal to Equation 174.

The procedure to determine the $MTTF_s$ for the logic solver in Figure A.14 is exactly the same. First determine the state differential equations.

$$(Eq. A.175) \quad P_1 = -[\theta + \mu_E] P_1(t) + [\lambda_1 + \lambda_2] P_2(t)$$

$$(Eq. A.176) \quad P_2 = -[\lambda_1 + \lambda_2] P_2(t) + \mu_E P_1(t)$$

$$(Eq. A.177) \quad P_0 = \theta P_1(t)$$

Again transferring Equations 175-177 into the Laplace domain using the same initial conditions (i.e., $P_2(0)=1$ and $P_1(0)=P_0(0)=0$), we have:

$$(Eq. A.178) \quad sP_2(s) - 1 = -[\lambda_1 + \lambda_2] P_2(s) + \mu_E P_1(s)$$

$$(Eq. A.179) \quad sP_1(s) = -[\theta + \mu_E] P_1(s) + [\lambda_1 + \lambda_2] P_2(s)$$

$$(Eq. A.180) \quad sP_0(s) = \theta P_1(s)$$

Solve Equations 178-180 for $P_1(s)$, $P_2(s)$ and $P_0(s)$, respectively.

$$(Eq. A.181) \quad P_2(s) = \frac{1 + P_1(s) \mu_E}{[s + \lambda_1 + \lambda_2]}$$

$$(Eq. A.182) \quad P_1(s) = \frac{[\lambda_1 + \lambda_2] P_2(s)}{[s + \theta + \mu_E]}$$

$$(Eq. A.183) \quad P_0(s) = \frac{\theta P_1(s)}{s}$$

Replace $P_1(s)$ in Equation 181 by Equation 182, and rewrite Equation 181, we have:

$$(Eq. A.184) \quad P_2(s) = \frac{s + \theta + \mu_E}{(s + \theta + \mu_E)(s + \lambda_1 + \lambda_2) - (\lambda_1 + \lambda_2)\mu_E}$$

The denominator of Equation 184 is again a quadratic function. Equation 184 can be written as:

$$(Eq. A.185) \quad P_2(s) = \frac{s + \theta + \mu_E}{(s - E_1)(s - E_2)}$$

Having solved for $P_2(s)$, Equations 182 and 183 may be solved for $P_1(s)$ and $P_0(s)$, respectively:

$$(Eq. A.186) \quad P_1(s) = \frac{\lambda_1 + \lambda_2}{(s - E_1)(s - E_2)}$$

$$(Eq. A.187) \quad P_0(s) = \frac{\theta(\lambda_1 + \lambda_2)}{s(s - E_1)(s - E_2)}$$

The product in the denominator, E_1E_2 , must be determined in the same manner as before. Rewrite Equation 187 in partial fractions form:

$$(Eq. A.188) \quad P_0(s) = \frac{\theta(\lambda_1 + \lambda_2)}{s(s - E_1)(s - E_2)} = \frac{A}{s} + \frac{B}{s - E_1} + \frac{C}{s - E_2}$$

Simplify Equation 188 to obtain:

$$(Eq. A.189) \quad \theta(\lambda_1 + \lambda_2) = A(s - E_1)(s - E_2) + Bs(s - E_2) + Cs(s - E_1)$$

Let $s=0$, Equation 189 reduces to,

$$(Eq. A.190) \quad \theta(\lambda_1 + \lambda_2) = AE_1E_2$$

In order to determine the value of A, transfer Equation 188 into the time domain and take the limiting condition as time approaches infinity.

$$(Eq. A.191) \quad P_0(t) = A + Be^{E_1t} + Ce^{E_2t}$$

As time approaches infinity, the exponential terms in the right hand side of 191 are eliminated.

$$(Eq. A.192) \quad P_0(0) = I = A$$

Again the limiting condition as time approaches infinity is the same as s approaches zero. Therefore, the value of A from Equation 192 can be replaced in Equation 190 to determine the product of the roots.

$$(Eq. A.193) \quad E_1 E_2 = \theta(\lambda_1 + \lambda_2)$$

The mean time to failure for the model in Figure A.14 is:

$$(Eq. A.194) \quad MTTF_s = \int_0^{\infty} R(t)dt = \lim_{s \rightarrow 0} R(s) = \lim_{s \rightarrow 0} [P_1(s) + P_2(s)]$$

where $P_1(s)$ and $P_2(s)$ are given by Equations 185 and 186. The mean time to failure is:

$$(Eq. A.195) \quad MTTF_s = \frac{\theta + \mu_E}{E_1 E_2} + \frac{\lambda_1 + \lambda_2}{E_1 E_2}$$

Replacing the denominator of Equation 160 with Equation 193, the $MTTF_s$ for the model in Figure A.14 can be written as:

$$(Eq. A.196) \quad MTTF_s = \frac{\theta + \mu_E + \lambda_1 + \lambda_2}{\theta(\lambda_1 + \lambda_2)}$$

As was stated earlier in this clause, in order to merge the two states, 1_D and 1_U , into one state, state 1, the effective repair rate, μ_E , must be determined. Since the two models are assumed to be equal (provided that μ_E is used in the merged state model), then the mean time to failure for both models must be the same. Hence, to solve for μ_E , equate the equations for $MTTF_s$, namely Equations 174 and 196.

$$(Eq. A.197) \quad MTTF_s = \frac{\theta(\mu_{OT} + \theta) + \theta\lambda_1 + \lambda_2(\mu_{OT} + \theta)}{\theta[\lambda_1\theta + \lambda_2(\mu_{OT} + \theta)]} = \frac{\theta + \mu_E + \lambda_1 + \lambda_2}{\theta(\lambda_1 + \lambda_2)}$$

Simplify Equation 197 to the following form:

$$(Eq. A.198) \quad \frac{(\lambda_1 + \lambda_2)[\theta(\mu_{OT} + \theta) + \theta\lambda_1 + \lambda_2(\mu_{OT} + \theta)]}{\lambda_1\theta + \lambda_2(\mu_{OT} + \theta)} = \mu_E + (\theta + \lambda_1 + \lambda_2)$$

Replace λ_1 and λ_2 with λC and $\lambda C(1-)$, respectively. Therefore, $\lambda_1 + \lambda_2 = \lambda$, and Equation 198 becomes:

$$(Eq. A.199) \quad \frac{\lambda[\theta(\mu_{OT} + \theta) + \theta\lambda C + \lambda(1-C)(\mu_{OT} + \theta)]}{\lambda C\theta + \lambda(1-C)(\mu_{OT} + \theta)} = \mu_E + (\theta + \lambda)$$

Solve Equation 199 for the effective repair rate.

$$(Eq. A.200) \quad \mu_E = \frac{\lambda[\theta(\mu_{OT} + \theta) + \theta\lambda C + \lambda(1-C)(\mu_{OT} + \theta)]}{\lambda[C\theta + (1-C)(\mu_{OT} + \theta)]} - (\theta + \lambda)$$

Simplify Equation 200 to reduce to the following form:

$$(Eq. A.201) \quad \mu_E = \frac{\theta(\mu_{OT} + \theta) + \lambda(\mu_{OT} + \theta - C\mu_{OT})}{\mu_{OT}(1-C) + \theta} - (\theta + \lambda)$$

Equation 201 can be further simplified to the form:

$$(Eq. A.202) \quad \mu_E = \frac{\theta(\mu_{OT} + \theta) + \lambda(\mu_{OT} + \theta - C\mu_{OT}) - (\theta + \lambda)[\mu_{OT}(1-C) + \theta]}{\mu_{OT}(1-C) + \theta}$$

Further simplifications reduce Equation 202 to its final form:

$$(Eq. A.203) \quad \mu_E = \frac{\theta C \mu_{OT}}{\mu_{OT}(1-c) + \theta}$$

Equation 203 must be true under two limiting conditions. These conditions are:

1. When there are no diagnostic functions available, **C=0**, then the effective repair rate must be zero, $\mu_E=0$, because all the failures are undetected and there is no repair (the mission time of the logic solver is the time between periodic on-line inspections).

Let **C=0** in Equation 203 and evaluate the equation. The result is that $\mu_E=0$.

2. When there are perfect diagnostic functions available, **C=1**, the effective repair rate should be the on-line repair rate, $\mu_E = \mu_{OT}$. This is true because all failures are detected.

Let **C=1** in Equation 203 and evaluate the equation. The result is that indeed $\mu_E = \mu_{OT}$.

NOTE Equation 203 should be used to evaluate the effective repair rate, μ_E , for all merged states and this rate should be included in the general formula for the MTTFS^S given by Equation 147. The generalized fail spurious Markov model is given in Figure A.10.

Annex B (informative) — Logic solver model input data

These tables represent the input data for logic solvers submitted by 7 logic solver suppliers that have a global presence in the process sector. The data provided is an average of the values submitted. It is submitted to serve users so they may benchmark values they are provided from other sources. The values used in user calculations should originate from the logic solver supplier.

This annex gives tables for:

- Hardware failure rates and common cause fractions
- Failure mode ratios
- Diagnostic coverage factors
- Systematic failures
- Repair time
- Configuration size data

B.1 Hardware failure rates

Table B.1 — Hardware failure rates

Item	Failure Rate failures/million hours		
	Low	Typical	High
Main Processor Board (memory, bus logic, communication)	12.00	25.00	50.00
Backup Control Unit			
I/O Processor/Common logic I/O module	2.50	5.00	10.00
Single Digital Input Circuit	2.50	5.00	10.00
Single Digital Output Circuit	0.10	0.20	0.40
Single Analog Input Circuit	0.10	0.20	0.40
Single Analog Output Circuit	0.05	0.10	0.20
Relay (industrial type)	0.25	0.50	1.00
Electromechanical Timer	0.20	0.50	2.00
Solid state: Input circuit	1.50	2.50	5.00
Solid state: Output circuit	0.10	0.20	0.40
Solid state: Logic gate	0.10	0.20	0.40
Solid state: Timer	0.01	0.10	0.20
Inherently fail-safe solid state: Input circuit	0.10	1.00	2.00
Inherently fail-safe solid state: Output circuit	0.05	0.10	0.20
Inherently fail-safe solid state: Logic gate	0.10	0.20	0.40
Inherently fail-safe solid state: Off delay timer	0.001	0.01	0.10
Analog Trip Amplifier	0.05	0.50	1.00
Power supply	0.20	0.40	0.80
	2.50	5.00	10.00
Common Cause Failures	Fractions		
Common Cause Factor -- β factor	0.005	0.01	0.05

B.2 Failure mode ratios

Table B.2 — Failure mode ratios

Item	% Safe Failures		
	Low	Typical	High
Main Processor Board (memory, bus logic, communication)	40	50	60
Backup Control Unit			
I/O Processor/ Common logic I/O module	40	50	60
Single Digital Input Circuit	40	50	60
Single Digital Output Circuit	25	50	75
Single Analog Input Circuit	25	50	75
Single Analog Output Circuit	25	50	75
Relay (Industrial Type)	25	50	75
Electromechanical Timer	50	75	90
Solid state: Input Circuit	30	50	70
Solid state: Output Circuit	25	50	75
Solid state: Logic Gate	25	50	75
Solid state: Timer	25	50	75
Inherently fail-safe solid state: Input Circuit	25	50	70
Inherently fail-safe solid state: Output Circuit	-	99.9	-
Inherently fail-safe solid state: Logic Gate	-	99.9	-
Inherently fail-safe solid state: Off delay timer	-	99.9	-
Analog Trip Amplifier	-	99.9	-
Power Supply	25	50	75
<i>Total Systematic Failures</i>	80	95	99
	20	50	60

B.3 Diagnostic coverage factors

Table B.3 — Diagnostic coverage factors

Item	Safe Failure	Percentage (%)		
	Dangerous Failure	Low	Typical	High
Main Processor Board (memory, bus logic, communication)	SF	80	90	99
	DF	70	80	99
Backup Control Unit	SF	80	90	99
	DF	70	80	99
I/O Processor/Common logic I/O module	SF	70	85	99
	DF	60	75	99
Single Digital Input Circuit	SF	0	50	99
	DF	0	25	99
Single Digital Output Circuit	SF	0	50	99
	DF	0	25	99
Single Analog Input Circuit	SF	0	50	99
	DF	0	25	99
Single Analog Output Circuit	SF	0	50	99
	DF	0	25	99
Relay (Industrial Type)	NA	NA	NA	NA
Electromechanical Timer	NA	NA	NA	NA
Solid state: Input Circuit	SF	0	50	99
	DF	0	25	99
Solid state: Output Circuit	SF	0	50	99
	DF	0	25	99
Solid state: Logic Gate	SF	0	50	99
	DF	0	25	99
Solid state: Timer	SF	0	50	99
	DF	0	25	99
Inherently fail-safe solid state: Input Circuit	SF	25	50	99.9
	DF	NA	NA	NA
Inherently fail-safe solid state: Output Circuit	SF	25	50	99.9
	DF	NA	NA	NA
Inherently fail-safe solid state: Logic Gate	SF	25	50	99.9
	DF	NA	NA	NA
Inherently fail-safe solid state: Off delay timer	SF	25	50	99.9
	DF	NA	NA	NA
Analog Input/Trip Amplifier	SF	0	50	99
	DF	0	25	99
Power Supply	SF	90	95	99
	DF	90	95	99

NA= Not Applicable

B.4 Systematic failures

Systematic failures⁽⁸⁾ demonstrate themselves completely different from random hardware failures. Random hardware failures can be caused by all kinds of stressors and the susceptibility of the components for these stressor(s). By careful testing before operation we assume that there are no hardware failures at the start of the operational period.

Systematic failures show a different behavior. During the design and engineering of the safety loops and software and (chip) hardware, failures arise that cannot be detected by tests. The numbers of systematic failures do not increase during the operational period, but can be triggered under all kind of “normal” operational conditions.

To calculate this particular behavior of systematic failures in the Markov model initialization values for the operational state, and **fail-dangerous** state has to be established for the three types of systematic failures in logic solvers.

The nature of the systematic failures is that they are present at the moment of the operational start of a logic solver. The probability that the logic solver is operational is < 100 % and the probability that the logic solver is not in the **fail-dangerous** state is > 0 %.

Table B.4.1 — Systematic failures — Initial probabilities

Item	Initial probability		
	Low	Typical	High
Engineering/Design Complexity	0.0001	0.001	0.01
Complex Chip hardware	0.0004	0.004	0.04
Software	0.00075	0.0075	0.075
Total Systematic Failures	0.00125	0.0125	0.125

By applying the appropriate techniques it is possible to detect systematic failures during the phases of the safety life cycle. The range of coverage factors applied to the systematic failures in the examples of Clause 6 are defined in table B.4.2.

Table B.4.2 — Systematic failures — Coverage factors

Item	Coverage factor		
	Low	Typical	High
Factory test	0.99	0.995	0.999
Engineering/Design Complexity	0.9	0.95	0.99
Complex Chip hardware	0.9	0.95	0.99
Software (embedded, utility, application)	0.9	0.95	0.99
Total Systematic Failures	0.9	0.95	0.99

The systematic failures are playing a major and, most times, a dominant role in safety instrumented systems and, as a possible alternative, the systematic failures in the table below indicate a broader range. In a logic solver comparison calculation that uses the table above the systematic failure range will in many cases dominate the results for the higher SIL. Using, as an example, a high Comparison Coverage factor ignores the dominant role and represents a better indication of the influence of the remaining parameters on the Safety Integrity Level. A practical range of the “Coverage factor factory test” will be, including systematic failures”:

Table B.4.3 — Systematic failures — Practical coverage factors factory test

Item	Coverage factor		
	Low	Typical	High
“Practical” Coverage factor factory test	0.9	0.925	0.95

To estimate the initial range for the PFD in the numeric Markov matrix methodology (as defined in ISA-TR84.00.02-2002 – Part 4), apply the diagnostic coverage factor for the factory test (defined in Table B.4.2) and assume a 50% probability that a systematic failure results in the dangerous state.

Table B.4.4 — Systematic failures — Start PFD

Item	Initial probability		
	Low	Typical	High
Start PFD	1.25E-05	6.25E-06	1.25E-06

For reliability calculations not applying the numeric Markov matrix methodology the systematic failures are also expressed in failure rates.

Table B.4.5 — Systematic failures — Failure rates

Item	Failure Rate: failures / million hours		
	Low	Typical	High
Engineering/Design Complexity	0.01	0.10	1.0
Complex Chip hardware	0.05	0.50	5.0
Software	0.09	0.90	9.0
Total Systematic Failures	0.15	1.50	15.0

B.5 Repair and test times

Table B.5 lists the values for the timing parameters used for the calculation examples given in Clause 6.

Table B.5 — Repair and test times

Item	Time in hours
	Typical
Repair time	
Applicable for: Detected Failures	8
Periodical Test Interval Time	
Time between "Functional Tests"	8760 (1 Yr.)

B.6 Configuration data

Table B.6 lists the number of items used for the calculation examples given in Clause 6.

Table B.6 — Configuration data

Item	Description	Config
	To define the physical size of the logic solver	
Electrical Systems		
n_{riv}	No. of Relays input voter	4
n_{rio}	No. of Relays logic	5
n_{ps}	No. of Power supplies	2
Electronic Systems		
n_i	No. of Input circuits	2
n_{oc}	No. of Output circuits	2
n_{giv}	No. of Logic gates input voter	4
n_{glo}	No. of Logic gates	5
n_{ps}	No. of Power supplies	2
Programmable Electronic Systems		
l	No. of Power supplies/leg	2
m	No. of DO modules/leg	2
n_{oc}	No. of Output circuits/DO module	2
n_{ic}	No. of Input circuits/DI module	2
Input measuring points		2
Output action points		2

Annex C — Index

accuracy	13, 60
alarm(s)	24
architecture(s)	9, 10, 18, 21, 22, 25, 27, 28, 30, 31, 39, 40
assessment	9
availability	11, 13, 49
boundary(ies)	12
calculation(s)	14, 21, 24, 25, 26, 27, 35, 36, 38, 39, 91, 96, 97
channel(s)	23, 24, 30, 31, 39, 54, 58, 59
closed	23, 56
common cause	11, 14, 15, 17, 22, 23, 31, 40
common cause failure(s)	14, 15, 22, 23
communication(s)	23, 24, 30, 31, 92, 93, 94
complex	13, 52, 68, 74
configuration(s)	9, 11, 14, 15, 17, 26, 27, 42, 43, 45, 46, 47, 51, 80
Configuration(s)	44
contact	39, 56
cost	14
coverage	9, 15, 17, 24, 28, 91, 95
coverage factor	28, 91, 95
covert	17
covert fault(s)	17
dangerous detected failure(s)	40, 52, 53, 56, 57, 58, 59, 60, 62
dangerous hardware failure	31
dangerous undetected failure rate	28, 40
dangerous undetected failure(s)	22, 23, 28, 31, 40, 52, 54, 55, 56, 57, 58, 59, 62
definitions	14, 73, 74

demand	9, 11, 13, 18, 22, 49, 56, 68, 69, 73
demand mode	11, 13
designer	9, 14
detected faults	22, 23
diagnostic coverage	9, 15, 17, 24, 28
diagnostic(s)	9, 15, 17, 20, 24, 28, 81, 82, 83, 90
diagram	27, 30, 39, 65
diversity	9, 13
document(s)	9, 11, 12, 13, 14, 17, 23, 60, 80
documents	14
fail-dangerous	28, 31, 34, 35, 44, 46, 49, 59, 62, 64
Fail-safe	22, 43, 45, 46, 47, 80
Failure Mode and Effect Analysis (FMEA)	19, 20, 49
failure mode(s)	17
failure rate data	14
failure rate(s)	14, 17, 19, 20, 24, 28, 33, 40, 42, 49, 52, 53, 62, 65, 81, 96
failure state(s)	55
false	14
fault tree(s)	13
field device(s)	9
final element(s) [See field device(s)]	11, 14, 23, 39
frequency	9, 13
function	15
function(s)	9, 10, 11, 13, 15, 21, 22, 24, 26, 28, 34, 35, 68, 69, 70, 71, 72, 73, 84, 85, 88, 90
functional test interval	21, 24, 26
functional test(s)	21, 24, 26
hardware	9, 13, 14, 20, 31, 39, 95, 96
hardware failure(s)	13, 20, 31, 95

hazard(s)	9
hazardous	14
hazardous event(s)	14
IEC	14
indicators	24
industry	9, 11
input module(s)	21, 23, 28, 54
inspection(s)	9, 13, 21, 52, 90
installation	11, 31
integration	80
interfaces	24
life cycle	11
logic solver(s)	11, 14, 15, 17, 18, 19, 20, 21, 22, 23, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 49, 50, 52, 54, 56, 57, 58, 59, 60, 61, 62, 64, 65, 66, 68, 69, 73, 74, 75, 76, 77, 78, 79, 80, 83, 85, 86, 90, 91, 95, 96, 98
maintenance	9, 10, 11, 13, 15
Markov analysis	9, 10, 14, 17, 18
Markov modeling	17, 18, 19, 25
matrix method(s)	96
measure(s)	11, 13
memory	92, 93, 94
mode(s)	11, 13, 17, 52, 54, 56, 59, 60, 91
modeling	14, 15, 17, 18, 19, 23, 25, 56, 60
modification(s)	13
MTTF ^{spurious}	10, 14, 24, 25, 49
nuisance trip	9, 14, 18
objective(s)	14, 18, 25
off-line	24, 80
on-line	20, 21, 24, 31, 60, 62, 69, 80, 81, 82, 83, 90

open	30
operator(s)	15, 56, 58
output(s) [See input/output devices and input/output modules]	21, 23, 24, 27, 28, 30, 31, 34, 35, 37, 38, 39, 43, 44, 45, 46, 47, 51, 54, 56, 58, 59
panel(s)	9
parameter(s)	9, 14, 15, 18, 62, 65, 71, 96, 97
period(s)	13, 14, 95
PFD_{avg}	10, 13, 14, 24, 25, 49
physical	98
plant	21
power	24, 30, 40, 78
power supply(ies)	24, 30, 40
Probability of Failure on Demand (PFD)	18
process industry(ies)	9, 11
program(s)	25
Programmable Electronic System(s) (PES)	9, 10, 14
purpose(s)	9, 86
quality	9, 13
quantitative	14
random hardware failure(s)	13, 95
read	79
redundancy	9, 13, 17, 24, 28, 31, 80
redundant	11, 23, 40
reference(s)	12
relay(s)	42, 43
reliability	9, 10, 13, 18, 26, 61, 62, 74, 86, 96
repair(s)	15, 19, 21, 24, 25, 49, 52, 53, 56, 57, 58, 59, 60, 61, 62, 65, 69, 74, 80, 82, 83, 86, 89, 90
risk assessment	9

risk reduction	11
risk(s)	9, 11
safe	14, 20, 21, 22, 23, 28, 30, 31, 37, 38, 39, 40, 43, 45, 47, 49, 50, 51, 52, 56, 57, 58, 62, 75, 80
safe state(s)	21, 22, 23, 30, 51, 57
safety availability	11, 13
safety function(s)	9, 11, 13, 22, 24
Safety Instrumented System(s) (SIS)	9, 10, 11, 12, 13, 14, 17, 18, 19, 23
safety integrity	11, 13, 14
Safety Integrity Level (SIL)	9, 10, 11, 27, 96
Safety Integrity Level (SIL) Evaluation Techniques	9, 10, 17
scope	17, 24
sensor(s) [See field device(s)]	11, 14, 23
separate(s)	30
separated	75
sequence(s) of failure(s)	56
sequencer(s) of failure(s)	56
shutdown	14, 19, 20, 22, 26, 27, 28, 30, 31, 49
SIL 1	12
SIL 2	12
SIL 3	12
simple	49, 68, 80, 82
simplicity	78
simplified equation(s)	13
SIS architecture	9, 10
SIS components	10
software	9, 13, 14, 15, 95
solid state	26, 27, 39, 44, 45, 46, 47, 92, 93, 94
solid state logic	27, 44, 45, 46, 47

spurious trip(s)	14, 24, 31, 49
supplier(s)	9, 24, 91
system analysis techniques	14
systematic failure(s)	14, 15, 17, 95, 96
team	9
terminology	18
Test Interval (TI)	17, 21, 24, 26, 27, 61
test(s)	17, 21, 24, 26, 27, 61, 95, 96
testing	9, 13, 24, 80, 95
time(s)	13, 14, 15, 19, 21, 26, 49, 56, 57, 59, 61, 62, 65, 66, 68, 69, 70, 71, 73, 74, 75, 76, 78, 79, 80, 82, 83, 85, 86, 88, 89, 90, 96, 97
timer(s)	92, 93, 94
TR84.00.02	9, 10, 11, 14, 18
trip(s)	9, 14, 18, 24, 31, 49
uncertainty analysis	26
variable(s)	68, 71, 85
vendor(s)	23, 24, 25
verification	17
voting	26, 39, 40, 41, 42, 43, 44, 45, 46, 47

Developing and promulgating sound consensus standards, recommended practices, and technical reports is one of ISA's primary goals. To achieve this goal the Standards and Practices Department relies on the technical expertise and efforts of volunteer committee members, chairmen and reviewers.

ISA is an American National Standards Institute (ANSI) accredited organization. ISA administers United States Technical Advisory Groups (USTAGs) and provides secretariat support for International Electrotechnical Commission (IEC) and International Organization for Standardization (ISO) committees that develop process measurement and control standards. To obtain additional information on the Society's standards program, please write:

ISA
Attn: Standards Department
67 Alexander Drive
P.O. Box 12277
Research Triangle Park, NC 27709

ISBN: 1-55617-806-9