Czech Technical University in Prague
Faculty of Information Technology
Department of Digital Design

Hierarchical Dependability Models Based on Markov Chains

by

Ing. Martin Kohlík

A dissertation thesis submitted to
the Faculty of Information Technology, Czech Technical University in Prague,
in partial fulfilment of the requirements for the degree of Doctor.

Dissertation degree study programme: Informatics

Prague, September 2015

Supervisor:
doc. Ing. Hana Kubátová, CSc.
Department of Digital Design
Faculty of Information Technology
Czech Technical University in Prague
Thákurova 9
160 00 Prague 6
Czech Republic

Abstract and contributions

This dissertation thesis deals with dependability models allowing calculating the rate of events leading to a hazard state – a situation, where safety of the modeled dependable system (e.g. railway station signaling and interlocking equipment, automotive systems, etc.) is violated, thus the system may cause material loss, serious injuries or casualties. Hierarchical dependability models based on multiple Markov chains are proposed. These models allow expressing multiple redundancies made at multiple levels of a system consisting of multiple cooperating blocks. The hazard rates of the blocks are calculated independently and, when combined, they are used to calculate the hazard rate of the whole system. The independent calculations are significantly faster than the calculation of a single model composed of all models of the blocks. The dissertation thesis shows a method of reducing Markov chains and using them to create hierarchical dependability models. Three example studies are used to demonstrate the advantages of the hierarchical dependability models. The models used in the first and the second study are related to railway interlocking equipment, the third case study is an actual system used to detect the presence of a train and eliminate the possibility of a train accident.

Keywords:

Fault tolerant systems, Hierarchical systems, Reliability, Reliability engineering, Railway safety.

Acknowledgements

First of all, I would like to express my gratitude to my dissertation thesis supervisor, Dr. Hana Kubátová. She has been a constant source of encouragement and insight during my research and helped me with numerous problems and professional advancements.

I would like to thank to Dr. Petr Fišer for his crucial comments and remarks helping me with the final part of my research.

Special thanks go to the staff of the Department of Digital Design, who maintained a pleasant and friendly environment for my research. I would like to express special thanks to the department management for providing most of the funding for my research. My research has also been partially supported by the Ministry of Education, Youth, and Sport of the Czech Republic under research program MSM 6840770014, by the Czech Science Foundation as project No. 102/09/1668, and by CTU student's grants SGS10/118/OHK3/1T/18, SGS11/090/OHK3/1T/18, SGS12/094/OHK3/1T/18, SGS13/101/OHK3/1T/18, SGS14/105/OHK3/1T/18, and MOBILITY 7AMB14SK177.

I would like to express thanks to my colleagues and friends, namely Mr. Filip Štěpánek, Mr. Martin Daňhel and Mr. Jaroslav Borecký, and others, for their valuable comments and support.

Finally, I give the following Czech-written thanks to my family members, friends and all others for their support during my studies.

Děkuji rodině, přátelům i všem ostatním, kteří při mě po celou dobu mého studia stáli a podporovali mě.

Contents

List of Figures

List of Tables

Abbreviations

2oo2	Two-out-of-two system
DFT	Dynamic Fault Tree
FPGA	Field-Programmable Gate Array
FT	Fault Tree
FTA	Fault Tree Analysis
GSPN	Generalized Stochastic Petri Net
ICU	Internal Checking Unit (of Track Circuit System)
MC	Markov Chain
MTTF	Mean Time To Failure
MDS	Modified Duplex System
NMR	N-Modular Redundancy
PN	Petri Net
RBD	Reliability Block Diagram
SIL	Safety Integrity Level
SPN	Stochastic Petri Net
SPS	Secondary Power Supply (of Track Circuit System)
TCS	Track Circuit System

Chapter 1

Introduction

1.1 Problem Statement and Motivation

Mission-critical systems with guaranteed levels of safety and reliability parameters are used in many applications (e.g. aviation, medicine, space missions, and railway applications, etc.) with serious impacts to people and environment in case of their failure.

Such systems are composed of blocks based on various types of hardware (e.g. multi-core and many-core systems, programmable hardware like FPGA, etc.). Due to heterogeneous structure and several types of possible faults in various architectures and technologies, a realistic model, which has to be a basis for necessary certifications of such systems, is mostly complicated.

Currently used dependability parameters calculations/predictions are performed mostly as a three-level top-down process:

1. *Failure modes, effects, (and criticality) analysis* [1], is mainly a qualitative analysis used to study problems that might arise from malfunctions of the systems. It determines, by failure mode analysis, the effect of each failure on system operation and identifies single failure points, that are critical to mission success or crew safety. It may also rank each failure according to the criticality category of failure effect and probability occurrence.
2. *(Several) components-based model(s)* (fault trees, block diagrams, etc) [1] are constructed (e.g. by using Fault Tree Analysis [2]). Complex undesired events are defined, resolved into its immediate causes until the elementary events/causes are identified.
3. *The prediction of the failure rates of the elementary events* is usually based on MIL-HDBK-217 [3], PRISM [4], RIAC 217Plus [5], or a similar model.

Both fault trees and block diagrams are based on a decomposition of a modeled system to several independent logical subsystems (components). The dependability of the system is based on dependabilities of the components. This hierarchical approach allows dependability parameters of large and complex systems to be calculated easily.

The main disadvantage of these components-based models is their inability to model online (self-)repairing capabilities of the systems (hot-swap modular systems, reconfigurable FPGAs, etc.). State-based models (Markov chains, Petri nets, etc.) are able to model these capabilities easily. The disadvantage of using Markov modeling techniques is state-explosion leading to difficulties in construction, and consequently leading to the inability to compute realistic values of dependability characteristics.

Therefore, the main aim of this thesis is to propose a simplified dependability model and methods for easier dependability parameters computations. These models are state-based, thus they are able to model (self-)repairing capabilities easily, and they can be used to create hierarchical dependability model of a system, and consequently they allow dependability parameters of large and complex systems to be calculated without the state-explosion issues.

The simplification of a model (called *reduction* in this thesis) is introduced in the first part of this thesis. The reduction is the key step allowing hierarchical dependability models to be built.

The reduced models are inexact, thus the main disadvantage of the proposed method is the inaccuracy of the resulting dependability parameters values in specific case systems. This issue is not critical, if the inexact results are proven as pessimistic. In other words, the real system must be no less safe than the system modeled by the inexact model(s). The reduced models are pessimistic, thus the results can be used as the guaranteed values of the calculated dependability parameters.

The hierarchical models using multiple linked models to reflect the structure of a system are presented in the second part of this thesis. Multi-level hierarchy may be used to describe each part of a heterogeneous structure independently. The hazard rates of the reduced lowest-level models are used in higher-level models, and so on, until a top-level model is also reduced and its hazard rate is used as the hazard rate of the whole system.

The proposed hierarchical models can be used to

- calculate the Safety Integrity Level (SIL) [6] and Mean Time To Failure (MTTF),
- determine, whether an event can be tolerated/omitted safely (its hazard rate is lower than a limit value specified by SIL),
- calculate hazard rates of systems containing heterogeneous structures and various types of possible faults.

Hierarchical models consisting of multiple small models

- are easier to read/understand,
- are easier to modify/manipulate,
- allow the exponential number of states of the model to be avoided, thus the dependability parameters are calculated significantly faster.

The proposed reduction method is demonstrated on two types of case study systems. The first type contains multiple (up to 17) identical dependable blocks configured as an

N-modular redundant system (NMR). Models of the internal block redundancy used in the study systems are used as dependability models of railway/subway interlocking equipment used in Czech Republic. The total hazard rate of this system and SIL value is calculated in this case. This type is also used to present the ability of the proposed reduction to trade off between accuracy and calculation time.

The first type of case study systems is also used to present the main disadvantage of the reduction – the inaccuracy – and its solution – the *partial reduction*. The accuracy of the partial reduction is significantly improved, but the system has to be replaced/repaired before a prescribed operation time (e.g. a period of a preventive maintenance) is reached. Due to the improvement of accuracy, the SIL classification can be significantly increased (by 2 or 3 levels, when the warranty period is ca. 20 or ca. 15 years respectively in the presented case studies).

The second type is a model of an actual modular system with hot-swap repair capability that utilizes the reduction of a three-level heterogeneous dependability model based on Markov chains and reliability block diagrams to calculate MTTF.

1.2 Contributions of the Thesis

The main aim of this thesis is to propose simplified dependability models and methods for easier dependability parameters computations. These models are able to model (self-)repairing capabilities and they can be used to create a hierarchical dependability model of a system, thus they allow dependability parameters of large and complex systems to be calculated without the state-explosion issues.

The method is especially designed to

- allow the heterogeneous hierarchical dependability models to be built. Heterogeneous models allow multiple types of dependability models to be used in a model of a complex system.
- allow the dependability parameters calculations to be performed significantly faster – even in the case of large complex systems, where the results of classical detailed models are practically unreachable due to state explosion.
- provide pessimistic solution, i.e. the real system must be no less safe than the system modeled by the proposed models. The solution can be strictly or partially pessimistic. The partially pessimistic solution leads to the guaranteed levels of dependability parameters at the expense of the reduced maximal allowed operational time of the system.

The models and methods are experimentally verified on complex systems based on real models of the railway interlocking equipment.

1.3 Structure of the Thesis

The thesis is organized into 5 chapters as follows:

1. *Introduction*: Describes the motivation behind the efforts together with the goals. There is also a list of contributions of this dissertation thesis.
2. *Background and State-of-the-Art*: Provides the theoretical background, surveys related dependability models and introduces the fault classification system.
3. *Dependability Model Reduction Method*: Describes the reduction method of absorbing Markov chains, the hierarchical models and their reduction.
4. *Case Studies and Their Results' Comparisons*: Contains an experimental verification of the presented method. Three hierarchical models based on real models of the railway interlocking equipment are used.
5. *Conclusions*: Summarizes the results of the research, suggests possible topics for further research, and concludes the thesis.

Chapter 2

Background and State-of-the-Art

2.1 The Threats to Dependability: Failures, Errors, Faults

This section is abstracted from Basic Concepts and Taxonomy of Dependable and Secure Computing [7].

2.1.1 Failures, Errors, Faults

Correct service is delivered when the service implements the system function. A service failure, often abbreviated to failure, is an event that occurs when the delivered service deviates from correct service. A service failure is a transition from correct service to incorrect service, i.e., to not implementing the system function.

The period of delivery of incorrect service is a service outage. The transition from incorrect service to correct service is a service restoration. The deviation from correct service may assume different forms that are called service failure modes and are ranked according to failure severities.

Since a service is a sequence of the system's external states, a service failure means that at least one (or more) external state of the system deviates from the correct service state. The deviation is called an error. The adjudged or hypothesized cause of an error is called a fault.

It is important to note that many errors do not reach the system's external state and cause a failure. A fault is active when it causes an error, otherwise it is dormant.

When the functional specification of a system includes a set of several functions, the failure of one or more of the services implementing the functions may leave the system in a degraded mode that still offers a subset of needed services to the user. The specification may identify several such modes, e.g., slow service, limited service, emergency service, etc. Here, we say that the system has suffered a partial failure of its functionality or performance.

Alternative definitions of failure, malfunction, faults, etc., used by International Electrotechnical Commission can be found in [8].

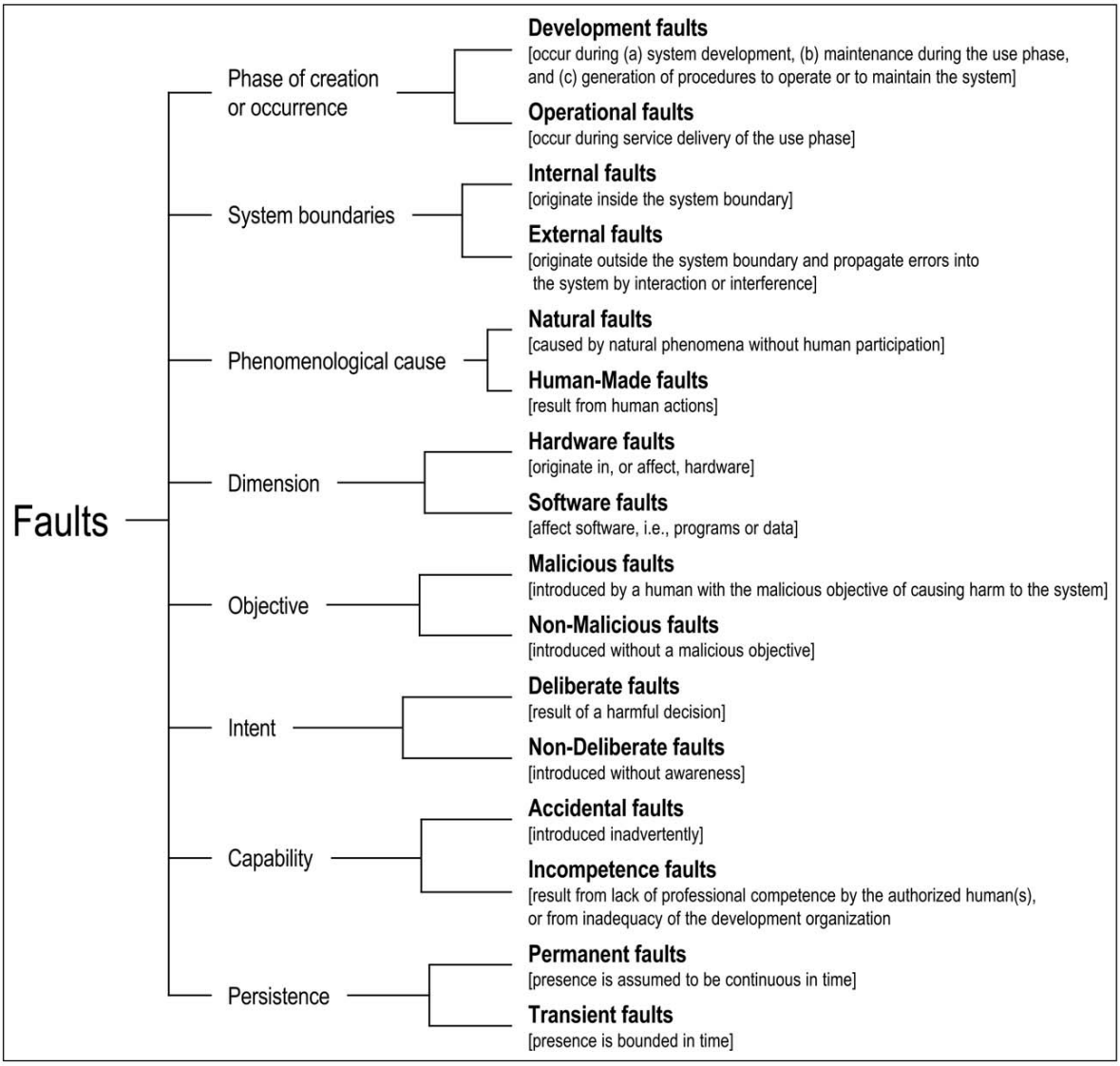

Figure 2.1: The elementary fault classes (taken from [7]).

2.1.2 Fault Classification

All faults that may affect a system during its life are classified according to eight basic viewpoints, leading to the elementary fault classes, as shown in Fig. 2.1.

If all combinations of the eight elementary fault classes were possible, there would be 256 different combined fault classes. However, not all criteria are applicable to all fault classes; for example, natural faults cannot be classified by objective, intent, and capability. 31 likely combinations are identified in [7].

The combined fault classes belong to three major partially overlapping groupings:

- *Development faults* that include all fault classes occurring during development.
- *Physical faults* that include all fault classes that affect hardware.
- *Interaction faults* that include all external faults.

Knowledge of all possible fault classes allows the user to decide which classes should be included in a dependability and security specification.

The models used in this thesis are focused on natural operational-time faults (both permanent and transient), but they are able to handle any type of fault.

Natural faults are physical (hardware) faults that are caused by natural phenomena without human participation. Production defects are natural faults that originate during development. During operation the natural faults are either internal, due to natural processes that cause physical deterioration, or external, due to natural processes that originate outside the system boundaries and cause physical interference by penetrating the hardware boundary of the system (radiation, etc.) or by entering via use interfaces (power transients, noisy input lines, etc.).

More details about the other fault classes (malicious and non-malicious human-made faults, development faults, etc.) can be found in [7].

2.2 Dependability Basics

Electronic reliability design handbook MIL-HDBK-338B [1] introduces the basic dependability terms clearly and comprehensibly. Therefore, Sections 2.2.1 – 2.2.3, and 2.2.5 are almost literally taken from this handbook.

Another well-written summary of reliability theory including illustrative examples and case studies can be found in [9]. Practical-oriented summary of reliability engineering including mechanical components systems, software systems, etc., can be found in [10].

2.2.1 Reliability

Reliability is defined in terms of probability, probabilistic parameters such as random variables, density functions, and distribution functions are utilized in the development of reliability theory. Reliability studies are concerned with both discrete and continuous random variables. An example of a discrete variable is the number of failures in a given interval of time. Examples of continuous random variables are the time from system installation to failure and the time between successive system failures.

The cumulative (failure) distribution function $F(t)$ is defined as the probability in a random trial that the random variable is not greater than t, or

$$F(t) = \int_{-\infty}^{t} f(t)\,dt$$

where $f(t)$ is the probability density function of the random variable, time to failure. $F(t)$ is termed the "unreliability function" when speaking of failure. It can be thought

of as representing the probability of failure prior to some time t. If the random variable is discrete, the integral is replaced by a summation. Since $F(t)$ is zero until $t = 0$, the integration can be from zero to t.

The reliability function, $R(t)$, or the probability of a device not failing prior to some time t, is given by

$$R(t) = 1 - F(t) = \int_t^\infty f(t)\,dt$$

The rate at which failures occur in the interval $t1$ to $t2$, the failure rate, $\lambda(t)$, is defined as the ratio of probability that failure occurs in the interval, given that it has not occurred prior to $t1$, the start of the interval, divided by the interval length. Thus,

$$\lambda(t) = \frac{R(t) - R(t + \Delta t)}{\Delta t\, R(t)}$$

where $t = t1$ and $t2 = t + \Delta t$. The hazard rate, $h(t)$, or instantaneous failure rate, is defined as the limit of the failure rate as the interval length approaches zero, or

$$h(t) = \frac{f(t)}{R(t)}$$

Only constant hazard rates are used in models presented in this thesis, thus a hazard rate will be denoted as λ.

Mean time to failure is nothing more than the expected value of time to failure and is derived from basic statistical theory as follows:

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

2.2.2 Maintainability

In reliability, one is concerned with designing a system to last as long as possible without failure; in maintainability, the emphasis is on designing a system so that a failure can be repaired as quickly as possible. The combination of high reliability and high maintainability results in high system availability (see Section 2.2.3).

Maintainability is a measure, how easily and rapidly a system or equipment can be restored to operational status following a failure. It depends on parameters given by the function of the equipment design and installation, personnel availability in the required skill levels, adequacy of maintenance procedures and test equipment, and the physical environment under which maintenance is performed.

As with reliability, maintainability parameters are also probabilistic and are analyzed by the use of continuous and discrete random variables, probabilistic parameters, and statistical distributions. An example of a discrete maintainability parameter is the number of maintenance actions completed in some time t, whereas an example of a continuous maintainability parameter is the time to complete a maintenance action.

2.2.3 Availability

The concept of availability was originally developed for repairable systems that are required to operate continuously, and are at any random point in time either operating or "down" because of failure and are being worked upon so as to restore their operation in minimum time. In this original concept a system is considered to be in only two possible states – operating or in repair – and availability is defined as the probability that a system is operating at any random point in time t, when subject to a sequence of "up" and "down" cycles which constitute an alternating renewal process. In other words, availability is a combination of reliability and maintainability parameters.

System availability can be defined in the following ways:

- *Instantaneous Availability* $A(t)$ – Probability that a system will be available for use at any random time t after the start of operation.
- *Mission Availability* $A_m(t_2 - t_1)$ – The proportion of time in an interval $(t_2 - t_1)$, during a mission, when a system is available for use, or

$$A_m(t_2 - t_1) = \frac{1}{t_2 - t_1} \int_{t_1}^{t_2} A(t)\, dt$$

  This is also called average availability A_{AV}.
- *Steady State of Availability* $A_S(t)$ – Probability a system will be available for use at a point in time t after the start of system operation as t becomes very large, or as $t \to \infty$, or

$$A_S = \lim_{t \to \infty} A(t)$$

2.2.4 Safety

Safety is the state of being "safe", the condition of being protected against physical, social, spiritual, financial, political, emotional, occupational, psychological, educational or other types or consequences of failure, damage, error, accidents, harm or any other event which could be considered non-desirable [11]. Safety can also be defined to be the control of recognized hazards to achieve an acceptable level of risk. This can take the form of being protected from the event or from exposure to something that causes health or economical losses. It can include protection of people or of possessions.

A target level of risk reduction in safety-critical systems (e.g. railway station signaling and interlocking equipment, automotive systems, etc.) is specified by SIL [6].

The value of SIL is calculated using the hazard rate of the system [6] and Table 2.1. E.g. the system classified as SIL4 is the only one safe enough to be used in the most critical applications, where hundreds or thousands of lives may be endangered by its failure.

Table 2.1: SIL value calculation table.

Hazard rate $\lambda\,[\mathbf{h^{-1}}]$	**SIL** [–]
$10^{-5} - 10^{-6}$	1
$10^{-6} - 10^{-7}$	2
$10^{-7} - 10^{-8}$	3
$10^{-8} - 10^{-9}$	4

2.2.5 Dependability Oriented Continuous Probability Distributions

Electronic reliability design handbook MIL-HDBK-338B [1] introduces several commonly used continuous distributions:

- *Exponential* – This is probably the most important distribution in reliability work and is used almost exclusively for reliability prediction of electronic equipment [3]. It describes the situation wherein the hazard rate is constant. The main advantages:
  - A single, easily estimated parameter (λ).
  - Has fairly wide applicability.
  - Is additive – that is, the sum of a number of independent exponentially distributed variables is exponentially distributed.
- *Gamma* – The gamma distribution is used in reliability analysis for cases where partial failures can exist, i.e. when a given number of partial failures must occur before an item fails (e.g. redundant systems) or the time to second failure when the time to failure is exponentially distributed.
- *Weibull* – The Weibull distribution is particularly useful in reliability work since it is a general distribution which, by adjustment of the distribution parameters, can be made to model a wide range of life distribution characteristics of different classes of engineered items.
- *Normal (Gaussian)* – There are two principal applications of the normal distribution to reliability. One application deals with the analysis of items which exhibit failure due to wear, such as mechanical devices. Another application is in the analysis of manufactured items and their ability to meet specifications. No two parts made to the same specification are exactly alike. The variability of parts leads to a variability in systems composed of those parts. The design must take this part variability into account, otherwise the system may not meet the specification requirement due to the combined effect of part variability. Another aspect of this application is in quality control procedures.
- *Lognormal* – The lognormal distribution is the distribution of a random variable whose natural logarithm is distributed normally; in other words, it is the normal

distribution with $\ln(t)$ as the variate. This is the most commonly used distribution in maintainability analysis. It applies to most maintenance tasks and repair actions comprised of several subsidiary tasks of unequal frequency and time duration.

Table shown in Fig. 2.2 taken from [1] shows the shapes of failure density, reliability and failure (hazard) rate functions for these distributions.

More details about probability distributions (both continuous and discrete), related probability functions, and probability theory can be found in [12].

2.3 Common Dependability Models

2.3.1 Markov Chains

Electronic reliability design handbook MIL-HDBK-338B [1] introduces Markov chain (MC) as follows:

Markov modeling processes are stochastic processes using random variables to describe the states of the process, transition probabilities for changes of state and time or event parameters for measuring the process. A stochastic process is said to be a Markov property if the conditional probability of any future event, given any past events and the present state, is independent of the past events and depends only on the present state of the process. The advantages for using Markov modeling methods include the flexibility in expressing dynamic system behavior. These types of behavior include:

- *Complex repair* – Situations consisting of repairs of either individual components or groups of components or partial repair of components.
- *Standby spares* – Standby conditions include hot, warm and cold spares. Hot spares are power-on units with identical stresses as apply to the active units, where warm spares have power-on but have lower stresses. Cold spares are power-off units.
- *Sequence dependency* – This behavior includes: functional dependency in which the failure of one component can cause the unavailability of other components; priority dependency in which behavior will differ depending on whether an event occurs before or after another; and sequence enforcement in which it is impossible for certain events to occur before others have occurred.
- *Imperfect fault coverage* – Imperfect fault coverage conditions arise when a dynamic reconfiguration process that is invoked in response to a fault or component failure has a chance of not being successful leading to system failure.

The disadvantages of using Markov modeling techniques include state-explosion leading to difficulties in construction, and consequently leading to the inability to compute realistic values of dependability characteristics. The state-explosion problem emerges especially when multiple models of the subsystems are composed to one model of the whole system. The composition is made by the Cartesian product of the states of all MCs composed,

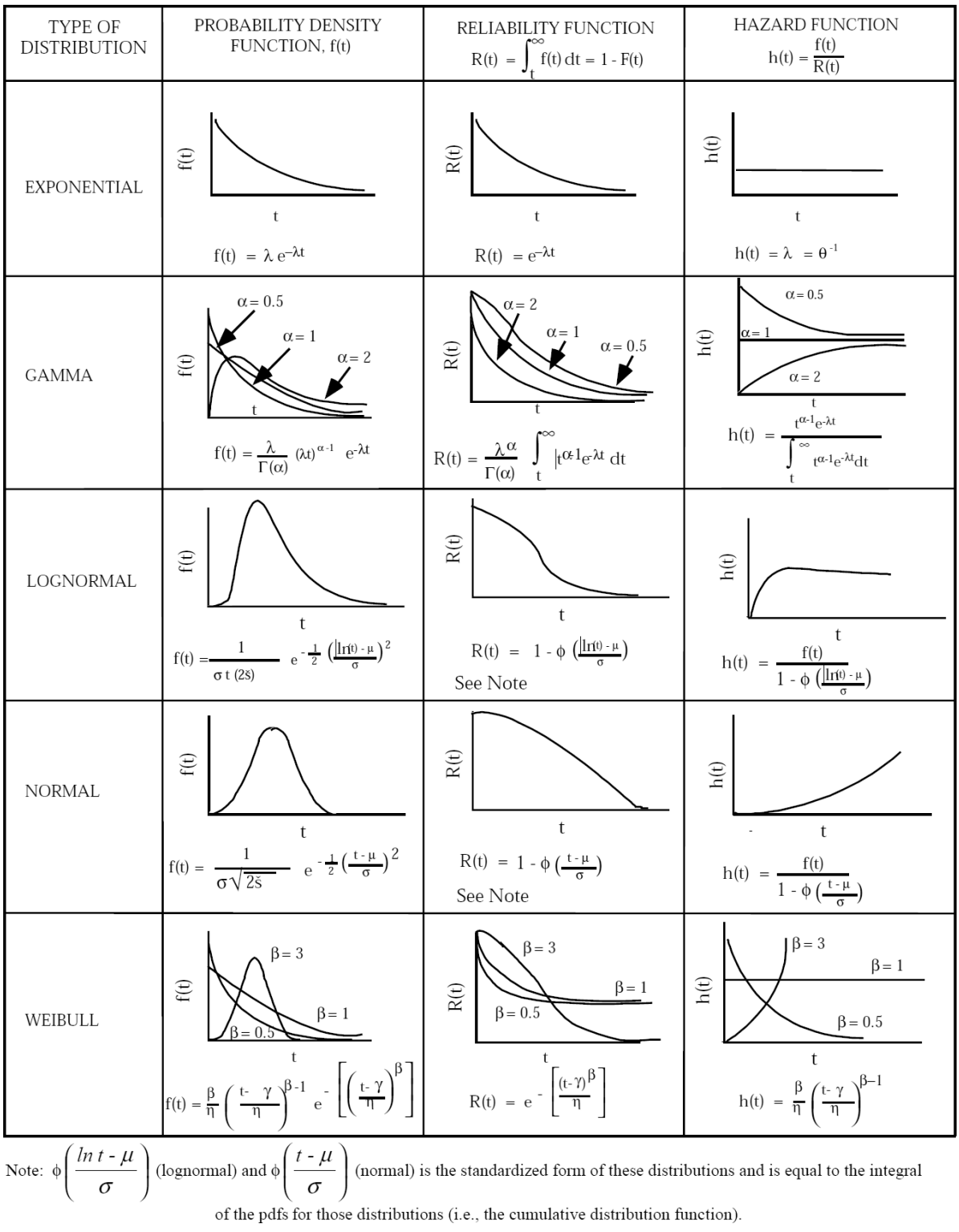

Figure 2.2: Shapes of failure density, reliability and failure (hazard) rate functions for commonly used continuous distributions (taken from [1]).

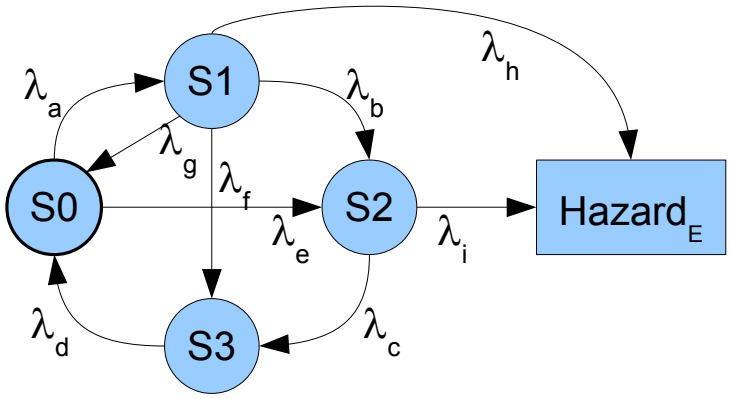

Figure 2.3: Illustrative example of absorbing Markov chain.

thus the number of the states of the MC of the system grows exponentially with increasing the number of models of the subsystems [9]. Solving models with thousands of states can challenge the computer resources available.

The composition using the Cartesian product can be avoided using a hierarchical composition of MCs. A hierarchical composition has been presented in [13]. This composition allows calculation of hazard rates of large systems using several MCs. The method uses equivalent models with an identical steady-state availability, but it is not concerned with the behavior during the operational time. This method cannot be applied on absorbing MCs (MCs containing states that, once entered, cannot be left), because steady-state availability of any absorbing MC is zero.

The illustrative example of an absorbing MC is shown in Fig. 2.3.

A system of differential equations is used to calculate the failure distribution function of the system modeled by the MC. The system contains one differential equation per state. The equation is constructed using the following template:

$$p'_{State}(t) = \sum_i (p_{Source\_State\_i}\, \lambda_{Hazard\_i}) - \sum_j (p_{State}\, \lambda_{Hazard\_j})$$

where i are indexes of the arcs leading to the *State*, $p_{Source\_State\_i}$ are the states, where the arcs origin, and j are indexes of the arcs leading from the *State*.

The template applied on the state *S2* of the illustrative example shown in Fig. 2.3 contain two arcs leading to the state *S2* and two arcs leading from the state *S2*:

$$p'_{S2}(t) = (p_{S1}\, \lambda_b + p_{S0}\, \lambda_e) - (p_{S2}\, \lambda_i + p_{S2}\, \lambda_c)$$

The equations of all states of the illustrative example follow:

$$p'_{S0}(t) = (p_{S3}\,\lambda_d + p_{S1}\,\lambda_g) - (p_{S0}\,\lambda_a + p_{S0}\,\lambda_e) \tag{2.1}$$
$$p'_{S1}(t) = p_{S0}\,\lambda_a - (p_{S1}\,\lambda_g + p_{S1}\,\lambda_b + p_{S1}\,\lambda_f + p_{S1}\,\lambda_h) \tag{2.2}$$
$$p'_{S2}(t) = (p_{S1}\,\lambda_b + p_{S0}\,\lambda_e) - (p_{S2}\,\lambda_i + p_{S2}\,\lambda_c) \tag{2.3}$$
$$p'_{S3}(t) = (p_{S2}\,\lambda_c + p_{S1}\,\lambda_f) - p_{S3}\,\lambda_d \tag{2.4}$$
$$p'_{Hazard_E}(t) = (p_{S1}\,\lambda_h + p_{S2}\,\lambda_i) \tag{2.5}$$

The second part of the system of the differential equations are the initial conditions. There is one initial condition per state. The initial probability of an initial state is 1, the initial probabilities of the other states are 0.

The initial conditions of the illustrative example follow:

$$p_{S0}(t) = 1 \tag{2.6}$$
$$p_{S1}(t) = 0 \tag{2.7}$$
$$p_{S2}(t) = 0 \tag{2.8}$$
$$p_{S3}(t) = 0 \tag{2.9}$$
$$p_{Hazard_E}(t) = 0 \tag{2.10}$$

The complete system of the differential equations of the illustrative example contains the equations (2.1)-(2.5) and the initial conditions (2.6)-(2.10).

The failure distribution function of the MC is the sum of probabilities of the hazard states.

The illustrative example contains only one hazard state, thus the failure distribution function $F(t)$ is

$$F(t) = p_{Hazard_E}(t)$$

2.3.2 Petri nets

Petri nets (PNs) are graphical and mathematical modeling tool applicable to many systems [14], [15], [16].

As a graphical tool, Petri nets can be used as a visual-communication aid similar to flow charts, block diagrams, and networks. In addition, tokens are used in these nets to simulate the dynamic and concurrent activities of the systems. An illustrative example of a Petri net is shown in Fig. 2.4.

As a mathematical tool, it is possible to set up state equations, algebraic equations, and other mathematical models governing the behavior of the system. The formal definition of a Petri net is a 5-tuple, $PN = (P, T, F, W, M_0)$ where:

$P = \{p_1, p_2, \ldots, p_m\}$ is a finite set of places,

$T = \{t_1, t_2, \ldots, t_n\}$ is a finite set of transitions,

$F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs (flow relation),

$W : F \to \{1, 2, 3, \ldots\}$ is a weight function,

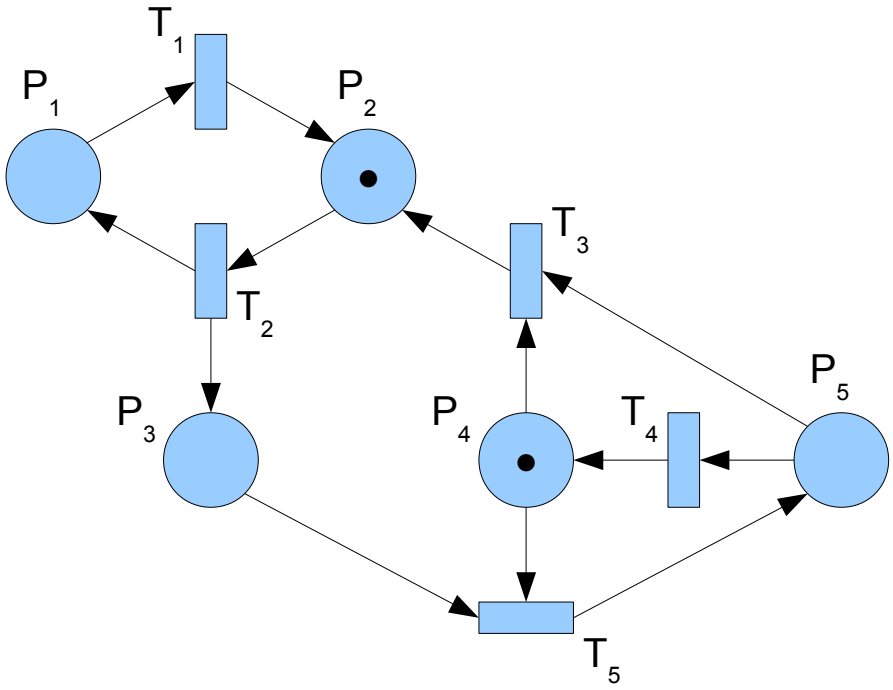

Figure 2.4: Illustrative example of Petri net (taken from [16]).

$M_0 : P \to \{0, 1, 2, 3, \dots\}$ is the initial marking,
$P \cap T = \emptyset$ and $P \cup T \neq \emptyset$.

A Petri net structure $N = (P, T, F, W)$ without any specific initial marking is denoted by N.

A Petri net with the given initial marking is denoted by (N, M_0).

The concept of time is not explicitly given in the original concept of the Petri net. However, for the performance evaluation of systems, it is necessary and useful to introduce time delay associated with transitions and/or places in their net models [16]. Such Petri net model is known as a (deterministic) timed net if the delays are deterministically given, or as a stochastic net (SPN) if the delays are probabilistically specified.

Generalized stochastic net (GSPN) [17] have two different classes of transitions: immediate transitions and (stochastic) timed transitions. Once enabled, immediate transitions fire in zero time. Timed transitions fire after a random, exponentially distributed enabling time as in the case of SPNs.

(G)SPNs have similar properties as MCs. A complex system may be modeled by a simple SPN, but the state explosion will occur in reliability function calculation, too.

(Finite size) MCs and (bounded) GSPNs can be mutually transformed. The following brief description is taken from [A.5]. The more detailed description can be found in [17].

The MC-to-SPN transformation is direct:

- Convert every Markov chain state into SPN place
- Convert every Markov chain edge into SPN transition (keep intensity rates)
- Add one token to the Petri net place created from the default state of the MC

The SPN-to-MC transformation is not as simple as the previous one. SPNs may have more than one token and they may contain transitions with multiple arcs. These two facts cannot be included in Markov chains. The transformation is based on the reachability graph of SPN. The reachability graph condenses each marking of SPN to one state and eliminates transitions with multiple arcs.

This transformation is made as follows:

- Create reachability graph of SPN
- Convert every reachability graph state into Markov chain state
- Convert every reachability graph edge into Markov chain edge (keep rates)

The GSPN-to-MC transformation is the most complicated one. It is based on the reachability graph of GSPN, too. The reachability graph of GSPN contains vanishing states (states with at least one immediate transition enabled), that cannot be included in Markov chains. The vanishing states of reachability graph must be removed. All edges leading to a removed vanishing state have to be connected to all edges starting from a removed vanishing state. The rates of edges have to be fixed as it is illustrated in Fig. 2.5.

The main issue of the (G)SPN-to-MC transformations is creating the reachability graph. The algorithm used to generate a reachability graph of a general (G)SPN is EXPSPACE-hard [18]. However, the algorithm generating specific cases of reachability graphs of (G)SPNs is NP-complete (live and safe free-choice PNs [19]) or even P (live t-systems [20]). This issue must be taken into account, if (G)SPN would be used in dependability modeling, the calculations of the dependability parameters using large and/or improper models can become computationally impossible in practice otherwise.

Despite this issue, (G)SPNs are still used as dependability models. A methodology to construct dependability models using GSPNs is described in [21], including an algorithm able to convert an FT into equivalent GSPN. (G)SPNs are used to model and analyze concurrent programs and architectures in [22].

2.3.3 Reliability Block Diagrams

Reliability block diagram (RBD) ([1], [23]) is a graphical analysis technique, which expresses the concerned system as connections of a number of components in accordance with their logical relation of reliability. Series connections represent logic AND of components, and parallel connections represent logic OR, while combinations of series and parallel connections represent voting logic. From the leftmost node to the rightmost node, there are several paths that are the conditions for successful operation of a system. If a component fails, the corresponding connection will be cut off. As failures of components occur,

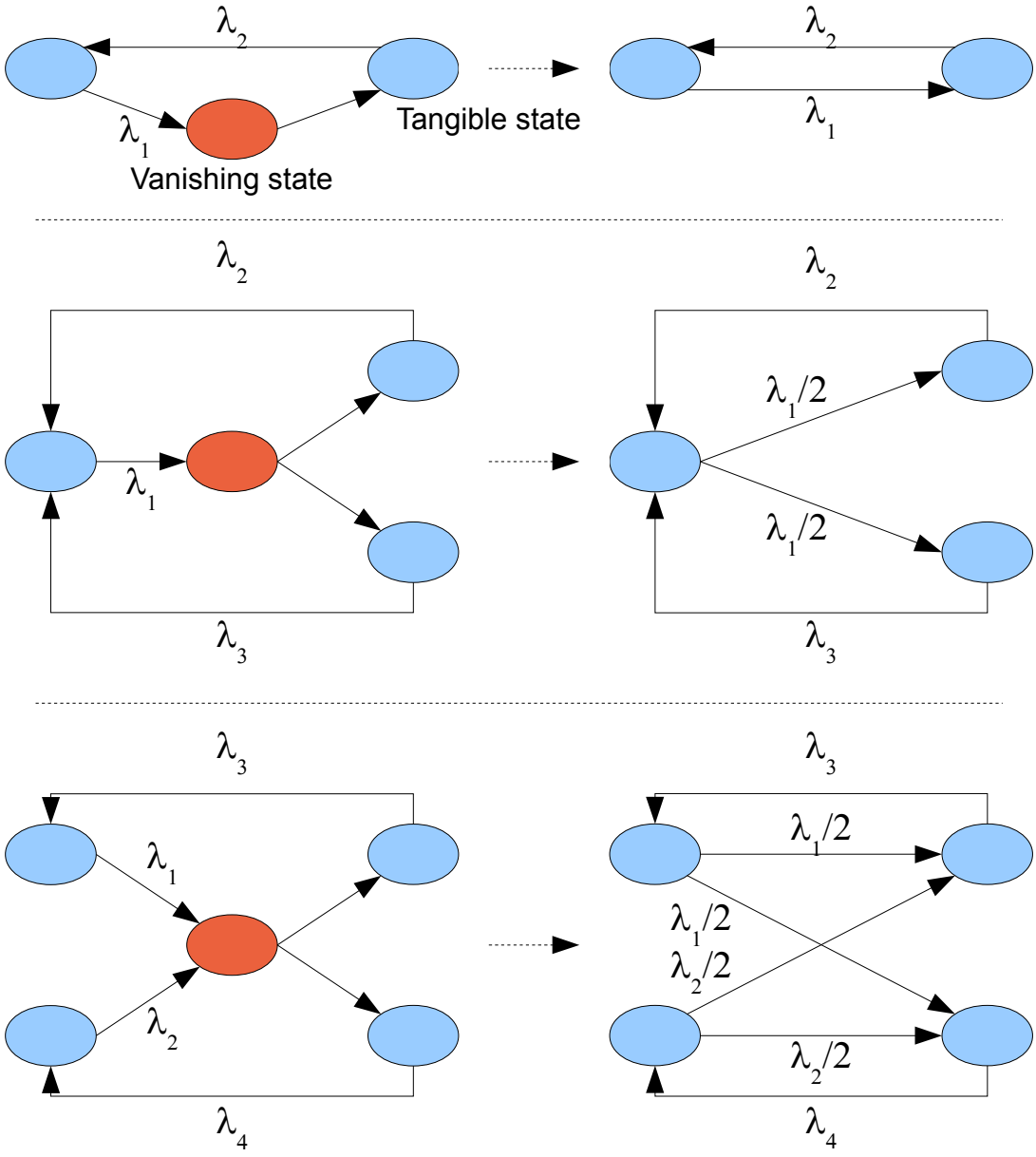

Figure 2.5: Illustrative examples showing the removal of the vanishing states of reachability graph of GPSN.

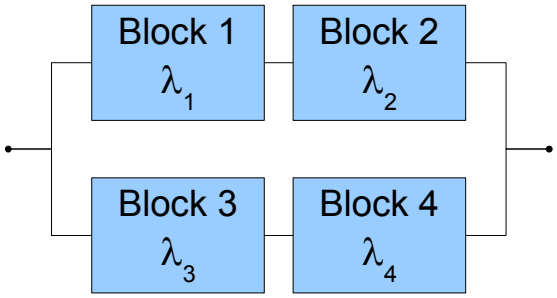

Figure 2.6: Illustrative example of reliability block diagram.

a system keeps operating successfully until no valid path from leftmost node to rightmost node can be made up of available connections. Then, the probability of failure of a system can be calculated according to probabilistic principles. The illustrative example of RBD is shown in Fig. 2.6.

The failure distribution function of a system modeled by the RBD is a combination of series and parallel functions, where

$$F_{Series}(t) = 1 - \prod_i (1 - F_i(t))$$
$$F_{Parallel}(t) = \prod_i (F_i(t))$$

The failure distribution of the illustrative example shown in Fig. 2.6 follows:

$$F_{12}(t) = F_1(t)\, F_2(t)$$
$$F_{34}(t) = F_3(t)\, F_4(t)$$
$$F(t) = 1 - ((1 - F_{12}(t))\,(1 - F_{34}(t)))$$

2.3.4 Fault Trees

This section is taken from Fault Tree Handbook with Aerospace Applications [2].

The Fault Tree (FT) is a graphic model of various parallel and sequential combinations of faults that will result in the occurrence of the predefined undesired event. The faults can be events that are associated with component hardware failures, human errors, software errors, or any other pertinent events which can lead to the undesired event. A fault tree thus depicts the logical interrelationships of basic events that lead to the undesired event, the top event of the fault tree.

It is important to understand that a fault tree is not a model of all possible system failures or all possible causes for system failure. A fault tree is tailored to its top event that corresponds to some particular system failure mode, and the fault tree thus includes only

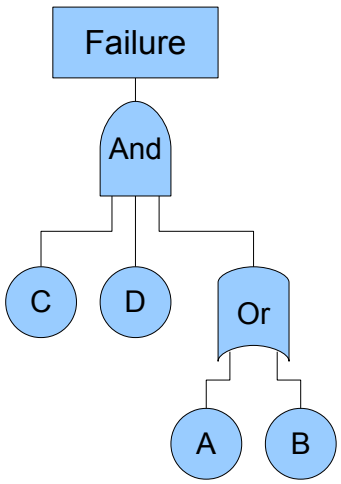

Figure 2.7: Illustrative example of fault tree.

those faults that contribute to this top event. Moreover, these faults are not exhaustive – they cover only the faults that are assessed to be realistic by the analyst.

It is also important to point out that a fault tree is not in itself a quantitative model. It is a qualitative model that can be evaluated quantitatively and it often is. This qualitative aspect, of course, is true for virtually all varieties of system models. The fact that a fault tree is a particularly convenient model to quantify does not change the qualitative nature of the model itself.

Intrinsic to a fault tree is the concept that an outcome is a binary event i.e., to either success or failure. A fault tree is composed of a complex of entities known as "gates" that serve to permit or inhibit the passage of fault logic up the tree. The gates show the relationships of events needed for the occurrence of a "higher" event. The "higher" event is the output of the gate, the "lower" events are the "inputs" to the gate. The gate symbol denotes the type of relationship of the input events required for the output event. Fig. 2.7 shows a simple fault tree containing four basic events and two gates. More types of events, gates, and other symbols can be found in [2].

2.3.4.1 Fault Tree Analysis

Fault Tree Analysis (FTA) is an application of deductive logic to produce an FT. Various failure modes that can contribute to a specified undesirable event are organized deductively and represented graphically. First the top undesired event is defined and drawn. Below this, secondary undesired events are drawn. These secondary undesired events include the potential hazards and failures that are immediate causes of the top event. Below each of these subevents are drawn second-level events, which are the immediate causes of the subevents. The process is continued until basic events are reached (often called elementary faults). Since the diagram branches out and there are more events at each lower level, it

resembles an inverted tree. The treelike structure of the diagram illustrates the various critical paths of subevents leading to the occurrence of the top undesired event.

Paper presented in [24] describes fault tree construction using FTA primary and secondary failure techniques as follows: A fault tree constructed using the primary failure technique is developed only to the point, where identifiable primary events will directly produce the required fault events. Construction using the secondary failure technique does not stop, when it reaches the component level. It continues until the effect on each component, of the possible failure of all related components, is identified.

2.3.4.2 Dynamic Fault Trees

Dynamic Fault Trees (DFTs) [2], [25], [26] contain several special types of gates that make them easier to model systems where the order in which events occur affects the outcome. These special gates are part of the DFT methodology that has been developed specifically for the analysis of computer-based systems. The concept of fault coverage is also introduced, which is used in the fault tolerant computing community to model a phenomenon similar to common-cause failures.

- Priority AND gate – The event is propagated to the output, if the input events come in the specified order.
- Functional dependency (FDEP) gate – Contains a single trigger event and several dependent basic events. The dependent basic events are functionally dependent on the trigger event. When the trigger event occurs, the dependent basic events are forced to occur (fail). The separate occurrence of any of the dependent basic events has no effect on the trigger event.
- Spare gate – The inputs to the spare gate are all basic events and are ordered. The first (usually drawn as leftmost) input is the primary event, while the second and subsequent inputs represent spares. The spare gate models the sequential activation of the spares: the first spare is activated when the primary fails; the second when the first fails, etc. The spare gate has one output that becomes true after all the input events occur.
- Coverage model "gate" [27] – This "gate" contains another model(s) – e.g. FT or RBD – that is used to calculate several possible outcomes using the occurrence of the fault as the entry point to the model. There are three possible outcomes – exits R, C, and S. The R or C exit is reached when a fault is covered, the S exit is reached when a fault is uncovered. Exit R from the coverage model represents transient restoration, the correct recognition of and recovery from a transient fault. Exit C from the coverage model represents permanent coverage, the determination of the permanent nature of the fault, and the successful isolation and (logical) removal of the faulty component. Exit S from the coverage model represents single point failure, in that a single fault causes the system to fail, generally when an undetected

error propagates through the system, or if the faulty unit cannot be isolated and the system cannot be reconfigured.

These advanced gates can overcome the main disadvantages of the regular FTs, but one issue still remains – (in)ability to model a system with complex repairs (e.g. reconfigurable FPGAs, hot-swap modular systems etc.) performed during its operational period. Coverage model "gate" can contain another model able to express a repair capabilities of the system, but this model must be able to provide a constant hazard rate for all possible outcomes.

2.3.5 Dependability Models – Summary

2.3.5.1 Component-based vs State-based Models

The component-based models (DFTs, RBDs, etc.) are greatly effective to model large complex systems easily thanks to their modularity [2]. Their disadvantage is an (in)ability to model a system with complex repairs performed during its operational period. Coverage model "gates" of DFTs are able to express repair capabilities of the system, but they must contain another (state-based) model able to provide constant hazard rates.

The state-based models (MCs, GSPNs, etc.) can be used to model complex repairs, standby spares, sequence dependency, and imperfect fault coverage naturally. The disadvantages of using these models include state size and model construction. The number of the states of these models grows exponentially with increasing number of models of the subsystems. Solving models with thousands of states can challenge the computer resources available.

2.3.5.2 Composition of Component-based and State-based Models

There are several possibilities to compose component-based and state-based models:

- Coverage model "gate" of DFTs can contain state-based model able to express repair capabilities of the system, but this model must be able to provide constant hazard rate for all possible outcomes [2].
- A decomposition of DFT to several submodels has been presented in [28]. The method also allows state-based models to be used as the elementary events/blocks of the DFT, but it does not provide a procedure how to calculate a constant hazard rate from the state-based model, even though the constant hazard rate of the elementary event/block is necessary.
- The resulting hazard rate of the component-based model can be used as the fault rate of the state-based model easily. Constant hazard rate is required for analytical solution of the model, the simulation-based solution has to be used otherwise. The simulation-based solution may be faster than the analytical one, but its result is always an approximation of the actual solution. Moreover, there is no guarantee that the simulation-based solution is pessimistic.

Chapter 3

Dependability Model Reduction Method

A method allowing dependability models to be simplified for easier dependability parameters computations is proposed in this thesis. The method has been introduced in [A.1], [A.7], and [A.8], the improvements of calculation time and accuracy has been introduced in [A.2] and the draft of the final version is accepted (major revision is required) in [A.4].

The simplified models are created using the approximation of a general absorbing MC by a simple one – the *reduction* – made by merging all non-hazard states of a general MC into a single state that is called *Operational* in this thesis (see Fig. 3.1). The merge is feasible, because there is no need to distinguish among the non-hazard states in the hazard rate calculation. The reduced model contains a new hazard rate λ_{Hazard} – the hazard rate substituting all hazard rates in the exact model.

3.1 Reduction Algorithm

The hazard rate of the reduced model is calculated as a pessimistic value meeting the condition called the **main requirement** in this thesis. The main requirement is met

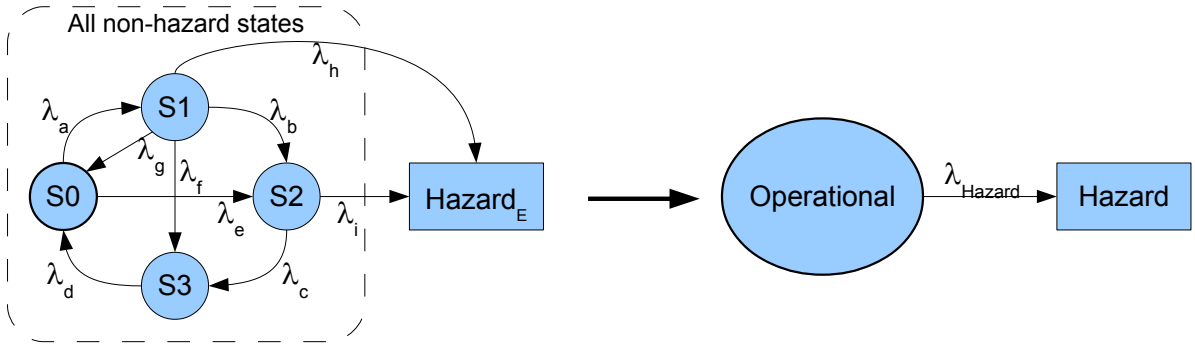

Figure 3.1: Illustrative example of dependability model reduction.

when

$$\forall t : F_R(t) \geq F_E(t)$$

where $F_E(t)$ is the failure distribution function of the non-reduced (exact) model and $F_R(t)$ is the failure distribution function of the reduced model – the solution of the following system of differential equations:

$$\begin{aligned} p'_{Operational}(t) &= -p_{Operational}(t)\,\lambda_{Hazard} \\ p'_{Hazard}(t) &= p_{Operational}(t)\,\lambda_{Hazard} \\ p_{Operational}(0) &= 1 \\ p_{Hazard}(0) &= 0 \end{aligned}$$

$$F_R(t) = p_{Hazard}(t) = 1 - e^{(-\lambda_{Hazard}\,.\,t)}$$

The reduction always leads to the same reduced model, thus these equations and the function $F_R(t)$ can be used for reduction of any model.

The drawback of the reduction is the loss of accuracy, because $F_E(t)$ can have any shape in general, while $F_R(t)$ has always an exponential shape, thus they are not equal.

There are two possibilities to perform the reduction:

- *Full reduction* – The main requirement is met strictly. This type of reduction leads to a strictly pessimistic solution, but it may be very inaccurate.
- *Partial reduction* – The main requirement is met only until the specified *limit* time value (t_{limit}). This way leads to a more accurate solution, but it can be used only when it is guaranteed that the preventive maintenance of the modeled system is performed (the system is replaced/repaired) before t_{limit} is reached. Three different ways to set the limit time value t_{limit} (the period of the preventive maintenance) have been introduced in [A.3]:
  - *Time-limited reduction* – uses t_{limit} itself
  - *Probability-limited reduction* – uses a p_{limit} probability
  - *Hazard-rate-limited reduction* – uses a λ_{limit} hazard rate

The full reduction is made by the algorithm shown below, the algorithm of the partial reduction is similar, but it uses $F_E(t)$ limited to interval $t \in [0; t_{limit}]$ (see the detailed description in Section 3.2).

The reduction process is:

1. Calculate the failure distribution function $F_E(t)$ by solving the system of differential equations corresponding to the exact model.

   The calculation of $F_E(t)$ is stopped, when the value of this function is differs from 1 by ϵ, where ϵ is extremely low value (10^{-10} is used in this thesis). This will allow us to limit $F_E(t)$ to interval $[0, t_\epsilon]$, where t_ϵ is calculated as $F_E(t_\epsilon) = 1 - \epsilon$.

$F_E(t)$ is sampled before the next step is performed. The sampling decreases the reduction time, but it may cause more inaccurate solutions. Samples are specified as pairs of values $[t, F_E(t)]$. The number of samples is specified by the *Samples per decade* parameter. The number of samples affects both the speed of the reduction and solution accuracy (see Section 4.4 for details).

2. Find the estimated value λ_{Hazard_Est}.

   The main goal of this step is to make a fast estimation of the hazard rate that will be used as the starting point for the next step. This value needs not meet the main requirement, but the better is the estimation, the faster will be the next step.

   A new estimated failure distribution function $F_{Est}(t) = 1 - e^{(-\lambda_{Hazard\_Est} \cdot t)}$ is derived in this step. This function is designed to intersect the failure distribution function $F_E(t)$ from the previous step at the time t_ϵ.

   $$F_{Est}(t_\epsilon) = F_E(t_\epsilon) = 1 - \epsilon$$

   Several estimated failure distribution functions can be used to obtain more precise estimation. λ_{Hazard_Est} is calculated from the most pessimistic (the highest) function.

3. Make correction of λ_{Hazard_Est} to satisfy the main requirement.

   The goal of the correction is to find a valid value of λ_{Hazard} – the lowest hazard rate whose $F_R(t)$ meets the main requirement with the required accuracy. The main requirement is tested on the sampled functions $F_E(t)$ and $F_R(t)$.

   The search is based on the bisection method (a method for iteratively converging to a solution which is known to lie inside some interval) searching for a point, where the λ_{Hazard} is valid (its $F_R(t)$ meets the main requirement). The maximal difference between $F_R(t)$ and $F_E(t)$ (the key value to test the main requirement) depends monotonically on the λ_{Hazard}, thus the bisection method can be used.

   The corrected hazard rate λ_{Hazard} is calculated as the estimated value λ_{Hazard_Est} multiplied by *result* value calculated using the algorithm shown in Fig 3.2.

   There are three main parts (see the flowchart shown in Fig. 3.2):

   a) find and verify the endpoints of the interval that will be bisected (*start* and *end*) – two loops.
      - *start* is an invalid hazard rate.
      - *end* is a valid hazard rate.

   b) Perform the bisection until the required accuracy given by *minStep* is met.

   c) Verify the result.

   The flowchart shown in Fig. 3.2 contains *verify value* subroutine checking whether all samples of $F_R(t)$ using selected *value* as hazard rate are greater than all samples of $F_E(t)$ or not. In other words, it checks whether the main requirement is met.

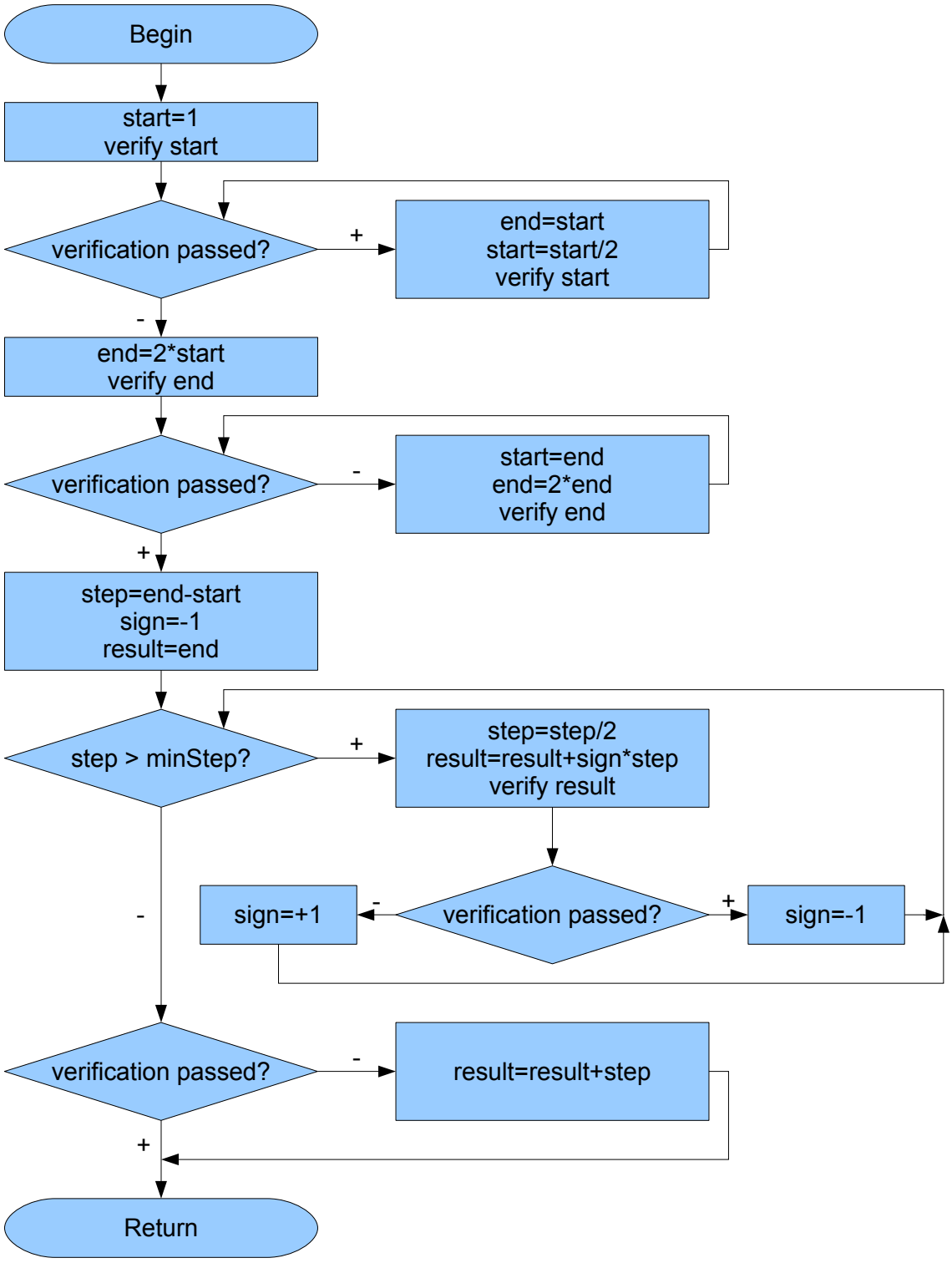

Figure 3.2: Correction algorithm flowchart.

There are two parameters that determine the trade-off between reduction accuracy and reduction time (see Section 4.4 for details):

- *minStep* – this parameter determines the end of the bisection loop of the correction. The error difference between minimal (ideal) hazard rate of the reduced model and λ_{Hazard} calculated by the reduction that is caused by non-zero *minStep* will be lower than *minStep*. *minStep* is expressed as a percentage of the estimated value λ_{Hazard_Est}.
- *Samples per decade* – this parameter determines the number of samples that will be tested for the main requirement fulfillment. It specifies the number of samples per each time decade measured in hours (i.e. in interval $\langle 10^n, 10^{n+1}) \, [h]$). The more samples are used, the better accuracy is achieved.

3.2 Partial Reduction Algorithm

Three ways to perform the partial reduction have been introduced in the previous section:

1. *Time-limited reduction* – uses t_{limit}

   This type of the partial reduction is very similar to the full one. The t_{limit} value is used instead of t_ϵ. The estimated function is designed to intersect the failure distribution function $F_E(t)$ at the time t_{limit}. The main requirement in the verification steps is checked in interval $[0, t_{limit}]$ only.

2. *Probability-limited reduction* – uses a p_{limit} probability

   Probability-limited reduction is time-limited reduction using t_{limit_p} limit value. t_{limit_p} value is calculated as $F_E(t_{limit\_p}) = p_{limit}$.

   The sampled version of $F_E(t)$ may not be defined in time t_{level_p}, because the samples are discrete, thus the time t_{level_p} may lay between two samples. If this situation happens, the closest sample beyond this value is used as the limit value to guarantee the validity of the main requirement in time t_{level_p}.

3. *Hazard-rate-limited reduction* – uses a λ_{limit} hazard rate

   The last type of partial reduction simply uses a λ_{limit} as a result. The limit value of the main requirement t_{limit_h} is calculated as $F_E(t_{limit\_h}) = 1 - e^{-\lambda_{limit} \cdot t_{limit\_h}}$.

   The time t_{level_h} may lay between two samples of $F_E(t)$. If this situation happens, the closest sample before this value is used as the limit, because that sample is the last one with the main requirement met.

3.3 Reduction Illustrative Example

The example is based on N-modular Redundancy (NMR) applied on 17 identical blocks and a voter. This voter is able to compare all outputs of the blocks. It uses majority voting

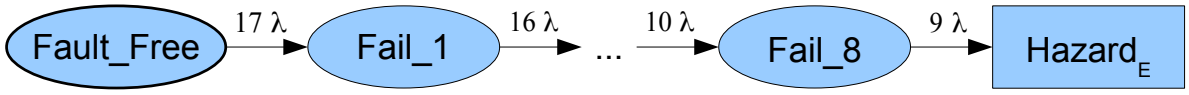

Figure 3.3: Dependability model of 17-modular redundant system.

to produce a single output. If less than half of the blocks fail, the voter is able to produce correct output. If more than half of the blocks fail, the voter will produce an incorrect output – this situation is considered as a hazard state. The erroneous blocks cannot be identified, thus there is no restoration/repair possibility.

Exact Dependability Model

The model shown in Fig. 3.3 is used to calculate the exact model failure distribution function of a generic NMR system. The NMR system containing 17 blocks will contain 8 transient states.

Dependability Model Reduction

The reduced model of the NMR block is the same as shown in the right part of the illustrative example in Fig. 3.1.

The steps of reduction correspond to the algorithm described in Section 3.1. The reduction parameters values are:

$$\lambda = 10^{-5}\,[h^{-1}]$$
$$minStep = 0.1\%$$
$$Samples\ per\ decade = 100$$

Calculate the failure distribution function $F_E(t)$ by solving the system of differential equations corresponding to the exact model.

The system of differential equations describing the dependability model of NMR containing 17 blocks is used for the calculation:

$$\begin{aligned}
p'_{Fault_Free}(t) &= -\,p_{Fault_Free}(t)\,17\lambda \\
p'_{Fail_1}(t) &= p_{Fault_Free}(t)\,17\lambda - p_{Fail_1}(t)\,16\lambda \\
p'_{Fail_2}(t) &= p_{Fail_1}(t)\,16\lambda - p_{Fail_2}(t)\,15\lambda \\
&\dots \\
p'_{Fail_8}(t) &= p_{Fail_7}(t)\,10\lambda - p_{Fail_8}(t)\,9\lambda \\
p'_{Hazard_E}(t) &= p_{Fail_8}(t)\,9\lambda \\
p_{Fault_Free}(0) &= 1 \\
p_{Fail_1}(0) &= p_{Fail_2}(0) = \dots = p_{Fail_8}(0) = p_{Hazard_E}(0) = 0
\end{aligned}$$

Find an estimated value λ_{Hazard_Est}

Table 3.1: The values of levels, times, when the exact function $F_E(t)$ crosses these levels, and the hazard rates λ_{Hazard_Est} of the reduction illustrative example system.

i	**ID$_i$**	**t$_{cross}$ [hours]**	**F$_E$(t$_{cross}$) [−]**	**λ_{Hazard_Est} [$\times 10^{-6}$ h^{-1}]**
1	212	128.8	0.379	1.884×10^{-18}
2	341	2,512	0.612	28.73×10^{-9}
3	423	16,596	0.759	0.0206
4	475	54,954	0.852	5.426
5	507	114,815	0.910	25.03
6	528	186,209	0.947	42.20
7	541	251,189	0.970	52.20
8	549	301,995	0.985	57.75
9	554	338,844	0.994	60.92
10	**557**	**363,078**	$\mathbf{1-\epsilon}$	**62.72** [1)]

[1)] The most pessimistic hazard rate value that is used as the estimated value in the correction step.

10 estimated failure distribution functions will be used in this example. Their intersections with the exact failure distribution function are given by

$$ID_i = \left\lceil \left(1.01 - 100^{-\left(\frac{i}{10}\right)}\right) . ID_\epsilon \right\rceil, \text{ where } i = 1 \ldots 10 \tag{3.1}$$

ID is the number of the sample. The sample ID_ϵ contains a pair $[t_\epsilon, F_E(t_\epsilon)]$.

This non-linear distribution is based on the experimental observations of failure distribution functions of exact dependability models. The observations indicate that the most pessimistic estimated failure distribution function intersects $F_E(t)$ in time close to t_ϵ in most cases. All values of the samples (ID, time, and the exact function $F_E(t)$ value), and the hazard rates λ_{Hazard_Est} of the estimated functions are shown in Table 3.1.

All 10 time intersection levels and estimations are shown in Fig. 3.4. The horizontal axis represents the time of operation measured in hours, the vertical axis represents the failure distribution function values. The black long-dashed line represents the exact model failure distribution function, the green line represents the most pessimistic estimated failure distribution function that is used in the correction step, and the red short-dashed lines represent the other estimated failure distribution functions. The horizontal and vertical lines show the intersections of the estimated and exact model failure distribution functions. The exact model failure distribution function is greater than the reduced model failure distribution function (i.e. the main requirement is not met) beyond these intersections.

Make correction of λ_{Hazard_Est} to satisfy the main requirement.

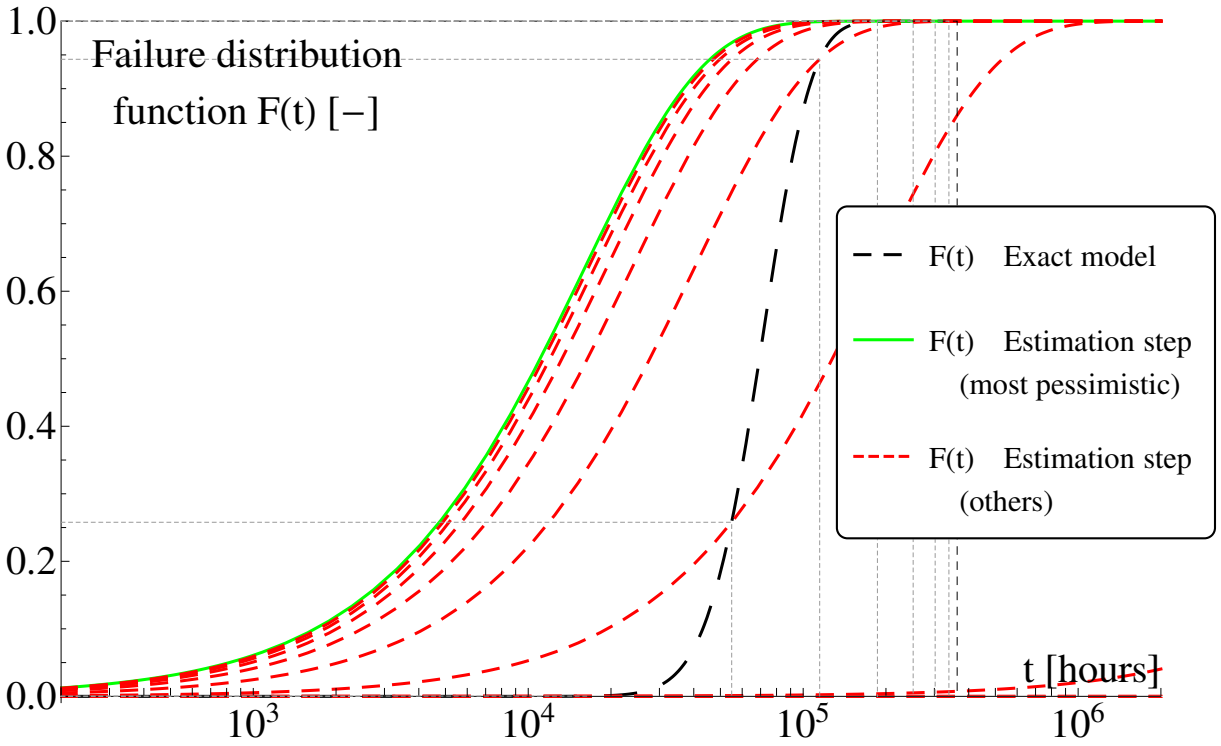

Figure 3.4: Failure distribution function of the exact model ($F_E(t)$) and estimated functions intersecting $F_E(t)$ at predefined time levels.

Table 3.2: The progress of the values of *start* and *end* variables during the first two iteration loops of the correction algorithm.

Loop	Iteration	Start value	End value	Verification result
1	1	1	–	Start passed
2	1	1	2	End failed
2	2	2	4	End passed

The actual algorithm would use the estimated value calculated in the previous step, but we will use the sample ID_5 taken from Table 3.1 to illustrate all parts (loops) of the correction algorithm (see the flowchart shown in Fig. 3.2 in Section 3.1).

The progress of the values of *start* and *end* variables during the first two iteration loops of the correction algorithm is shown in Table 3.2. As you can see, the values of *start* and *end* have been calculated in three steps and the value of *result* will be between 2 and 4 in this case.

The progress of the values of *result* and *step* variables during the main iteration loop of the correction algorithm is shown in Table 3.3. As you can see, the iteration runs while the value of *step* is not lower than *minStep* parameter (0.1%). If the verification passes,

Table 3.3: The progress of the internal variables during the first two iteration loops of the correction algorithm.

Iteration	Result		Step		Verification result
1	3	3.000	1	100 %	True
2	5/2	2.500	1/2	50 %	False
3	11/4	2.750	1/4	25 %	True
4	21/8	2.625	1/8	12.5 %	True
5	41/16	2.563	1/16	6.25 %	True
6	81/32	2.531	1/32	3.13 %	True
7	161/64	2.516	1/64	1.56 %	True
8	321/128	2.508	1/128	0.78 %	True
9	641/256	2.504	1/256	0.39 %	False
10	1,283/512	2.506	1/512	0.20 %	True
11	2,565/1,024	2.505	1/1,024	0.098 % [1)]	False [2)]
– [2)]	1,283/512	2.506	–	–	–

[1)] The current step is lower than the minStep – loop terminated.
[2)] The final verification of the loop failed, the last correction step is necessary.

the *result* value is decreased, if the verification fails, the *result* value is increased. The final correction is necessary, because the verification of the last iteration fails.

The result of the correction is

$$\lambda_{Hazard} = result \times \lambda_{Hazard\_Est} = 2.506 \times 25.03 \times 10^{-6} = 62.72 \times 10^{-6}\,[h^{-1}]$$

The plot shown in Fig. 3.5 shows the exact model failure distribution function, the estimated failure distribution function from the previous step, and the reduced model failure distribution function. The axes are identical to the axes used in the previous plot. The black long-dashed line represents the exact model failure distribution function, the red short-dashed line represents the estimated failure distribution function, and the green line represents the reduced model failure distribution function. The area, where the main requirement is not met, is highlighted by a light-gray shading (see the zoom window).

3.4 Partial Reduction Illustrative Example

The partial reduction illustrative example is based on the same system as the full reduction one (NMR17). The exact dependability model and the reduction parameters are identical.

- *Time-limited reduction* – using $t_{limit} = 60{,}000\,hours = 6.85\,years$

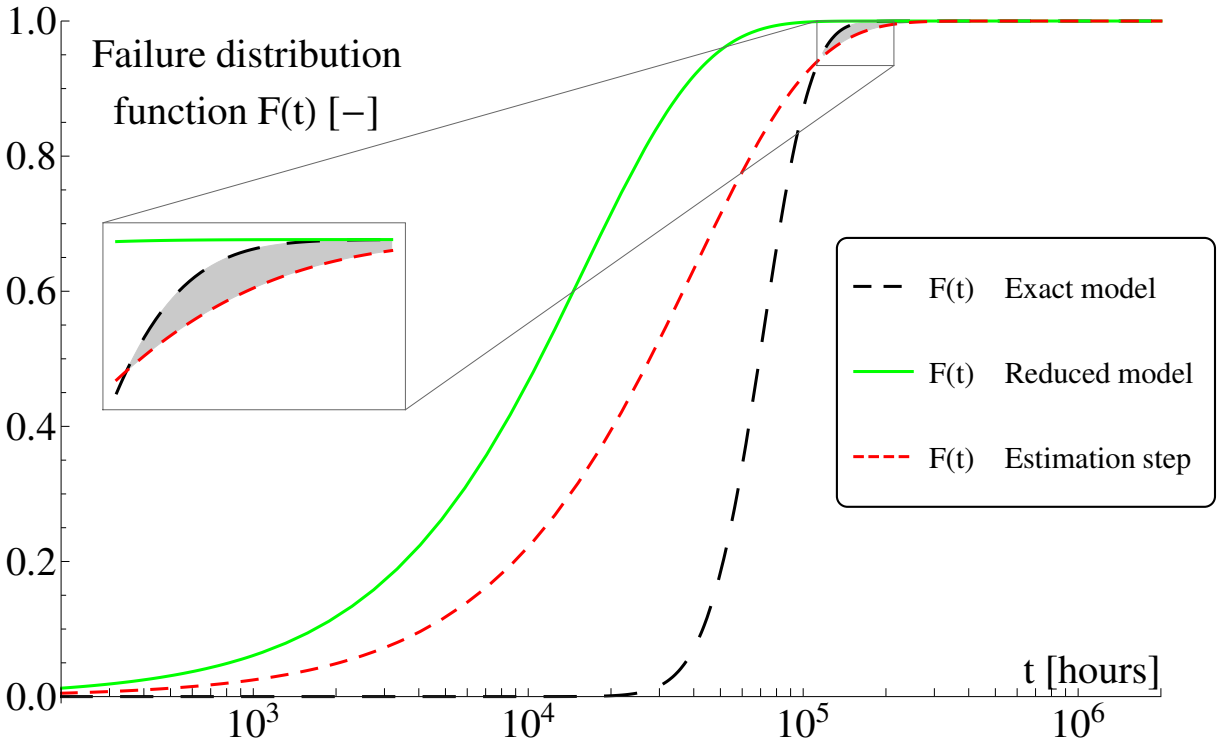

Figure 3.5: Estimated failure distribution functions and failure distribution functions of exact and reduced model.

The ID of the first sample beyond this value is

$$ID_{limit} = 479$$

All other ID_i values are calculated using Equation (3.1) shown in Section 3.3. ID_ϵ is replaced with ID_{limit}

All intersection levels and estimations are shown in Fig. 3.6. The meaning of the axes and lines is identical to the axes and lines of the plot shown in Fig. 3.4.

The plot shown in Fig. 3.7 shows the exact model failure distribution function, the full and the partial reduced model failure distribution function. The meaning of the axes is identical to the axes of the previous plot. The vertical line is the reduction limit t_{limit}.

The result of the time-limited reduction is

$$\lambda_{Hazard} = 7.041 \times 10^{-6}\,[h^{-1}]$$

- *Probability-limited reduction* – using $p_{limit} = 0.6$

  $t_{limit\_p} = F_E^{-1}(0.6) = 75{,}858\,[hours]\,(= 8.66\,years)$

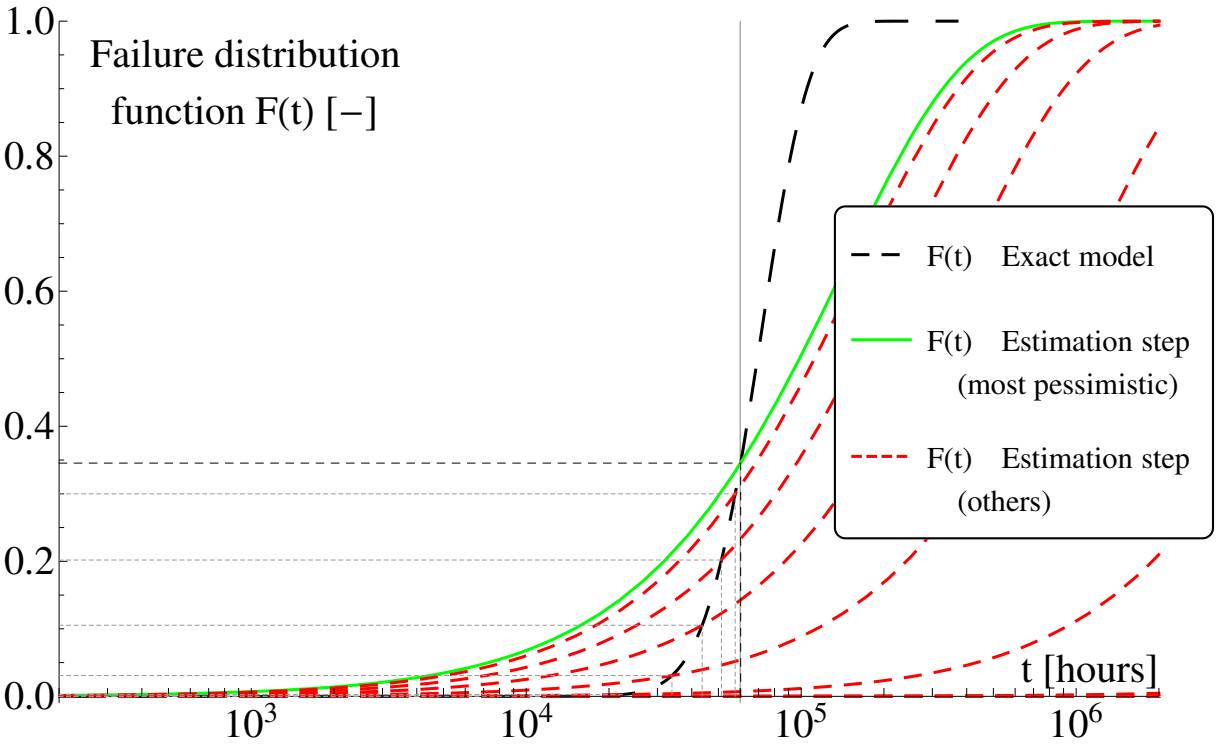

Figure 3.6: Failure distribution function of the exact model ($F_E(t)$) and estimated functions intersecting $F_E(t)$ at predefined probability levels (time-limited reduction using reduction limit $t_{limit} = 60{,}000\,hours$).

The rest of the probability-limited is similar to the time-limited reduction.

The plot shown in Fig. 3.8 shows the exact model failure distribution function, the full and the partial reduced model failure distribution function. The meaning of the axes is identical to the axes of the previous plot. The horizontal line is the reduction limit p_{limit}.

The hazard rate of the system reduced using probability-limited reduction is

$$\lambda_{Hazard} = 12.24 \times 10^{-6}\,[h^{-1}]$$

The period of the preventive maintenance of the system is

$$t_{limit} = 75{,}858\,hours = 8.66\,years$$

- *Hazard-rate-limited reduction* – using $\lambda_{limit} = 10^{-6}\,[h^{-1}]$

  The time of the first sample before the intersection of F_E and F_R is

  $$F_E(t_{limit\_h}) = F_R(t_{limit\_h}) \text{ iff } t_{limit\_h} = 33{,}884\,[hours]\,(= 3.87\,years)$$

  The period of the preventive maintenance of the system must be lower than this time.

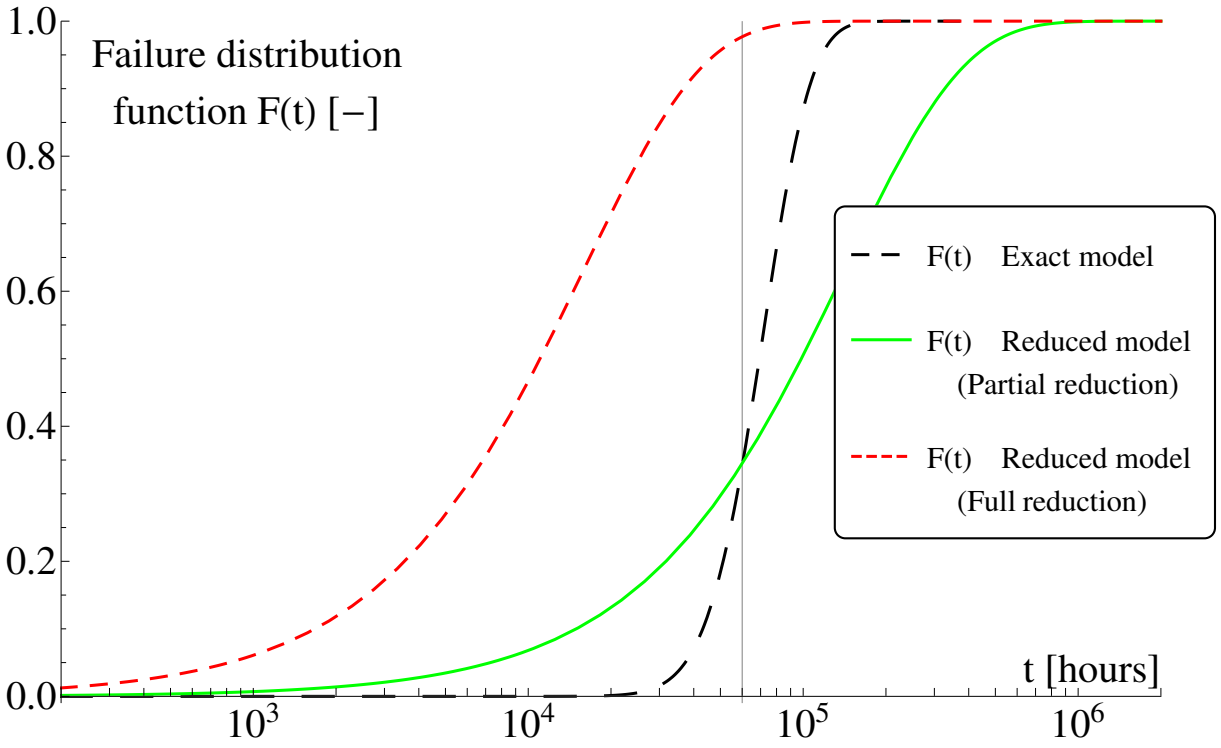

Figure 3.7: Comparison of failure distribution functions of exact and reduced model using full and partial reduction using reduction limit $t_{limit} = 60{,}000\,hours$.

3.5 Hierarchical Dependability Model and Reduction

The main principle of the reduction of the hierarchical model is to perform the reduction of the low-level dependability model(s), take the calculated hazard rate(s) and use it(them) in the upper level(s) of the hierarchy. An illustrative example of a two-level hierarchical model is shown in Fig. 3.9.

The current implementation assumes that all reductions use the same parameters (*minStep* and *Samples per decade*), but it is not mandatory.

The partial reduction of the hierarchical models requires the same t_{limit} value for all models used in the hierarchy. If the values were different, the partial reduction would be performed as if only the lowest limit was used. The partial reduction of the hierarchical models is performed as follows

1. *Time-limited reduction*

   The t_{limit} value is simply applied to all models – no further modifications are required.

2. *Probability-limited reduction*

   The inverse function $F_E^{-1}(p_{limit})$ cannot be used directly in this case, because $F_E(t)$ of the top-level model depends on the hazard rates of lower-level models.

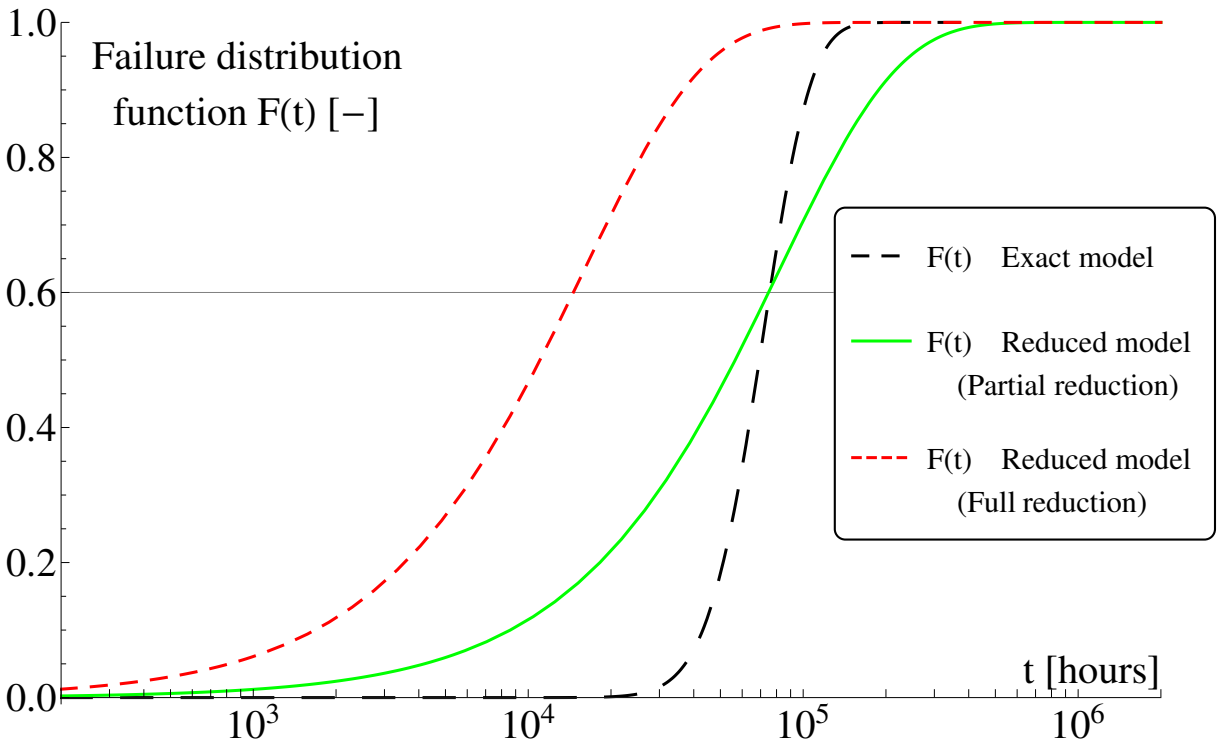

Figure 3.8: Comparison of failure distribution functions of exact model, and reduced model using full and partial reduction using reduction limit $p_{limit} = 0.6$.

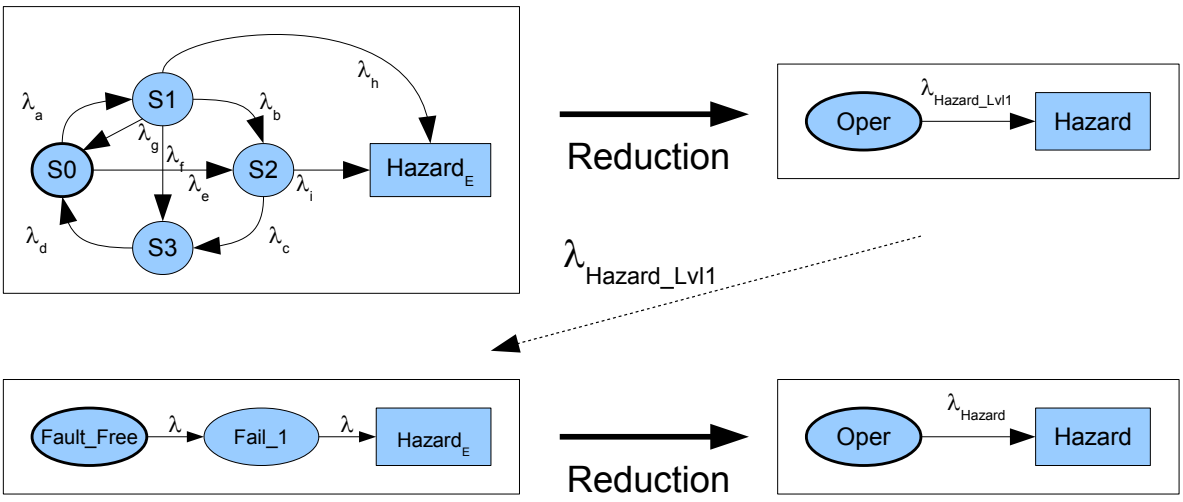

Figure 3.9: Illustrative example of the hierarchical dependability model.

The simple algorithm presented in [A.2] applies the full reduction on all lower-level models and the partial reduction on the top-level model.

The advanced algorithm allowing the partial reduction with the same t_{limit} value to be applied on all models used in the hierarchy is proposed in this thesis. The algorithm is based on bisection algorithm (see the flowchart shown in Fig 3.10).

The first cycle is used to set the boundaries of the bisection. The first value is taken from the probability-limited reduction of the low-level model. The time-limited reduction of the hierarchy is performed and the value $F_R(t_{limit})$ is calculated. If this value is lower than the p_{limit} value, the t_{limit} value is taken as the t_{start} value of the bisection and the value $t_{limit} \cdot 10$ is used in the next step. If $F_R(t_{limit})$ is greater than the p_{limit} value, the t_{limit} value is taken as the t_{end} value of the bisection and the value $t_{limit}/10$ is used in the next step. This step is performed until both t_{start} and t_{end} values are set.

The indexes of the samples at t_{start} and t_{end} are used as the boundaries of the bisection performed in the second cycle. The bisection is based on the time-limited reduction limited by the time of the sample laying between t_{start} and t_{end}. The higher is the used time limit, the higher is the $F_R(t_{limit})$ that is compared to the p_{limit}. The bisection ends, when the time step is lower than the difference between two closest samples.

The final correction is necessary, if the verification of the last iteration fails ($F_R(t_{limit})$ is lower than p_{limit}).

3. *Hazard-rate-limited reduction*

   The last type of partial reduction is similar to the probability-limited one. It is based on bisection, too. It uses the same boundaries and the time-limited reduction is performed in each iteration as well. The bisection is stopped when the the same condition is met. The only differences are the main condition ($\lambda_{result} < \lambda_{limit}$ is used instead of $F_R(t_{limit}) < p_{limit}$), the initial value of the first probability-limited reduction, and the last step preformed using the algorithm shown in Fig 3.11. The last step is different from the probability-limited reduction, because hazard-rate-limited reduction looks for a λ_{result} lower (or equal) than the limit value λ_{limit}.

The proposed reduction also allows a heterogeneous dependability model to be built. Such model may be based on (G)SPN, RBD, or (D)FT and includes a part modeled by an MC. An MC is reduced in such case and the constant hazard rate can be used in the base model.

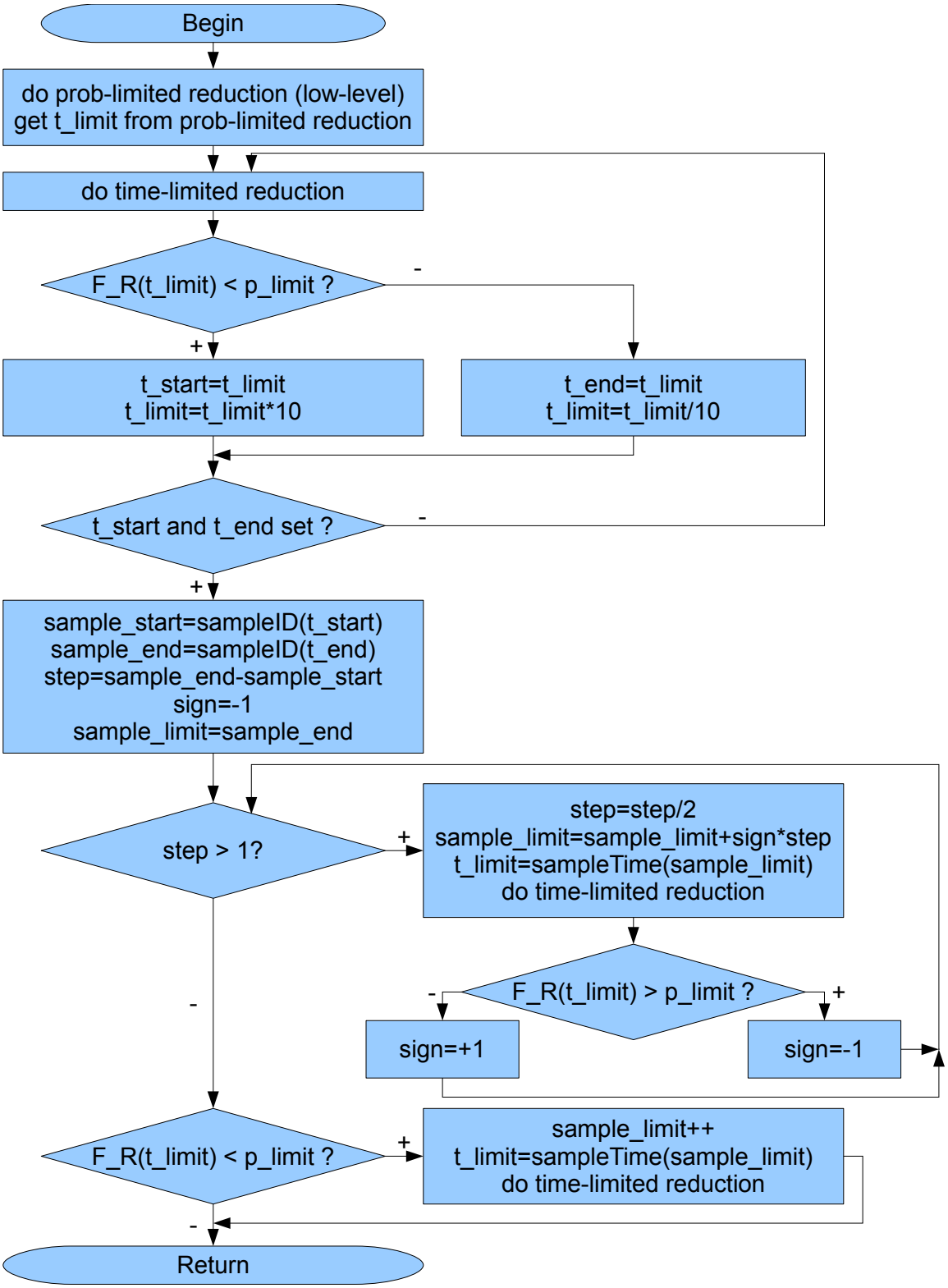

Figure 3.10: Probability-limited reduction algorithm flowchart.

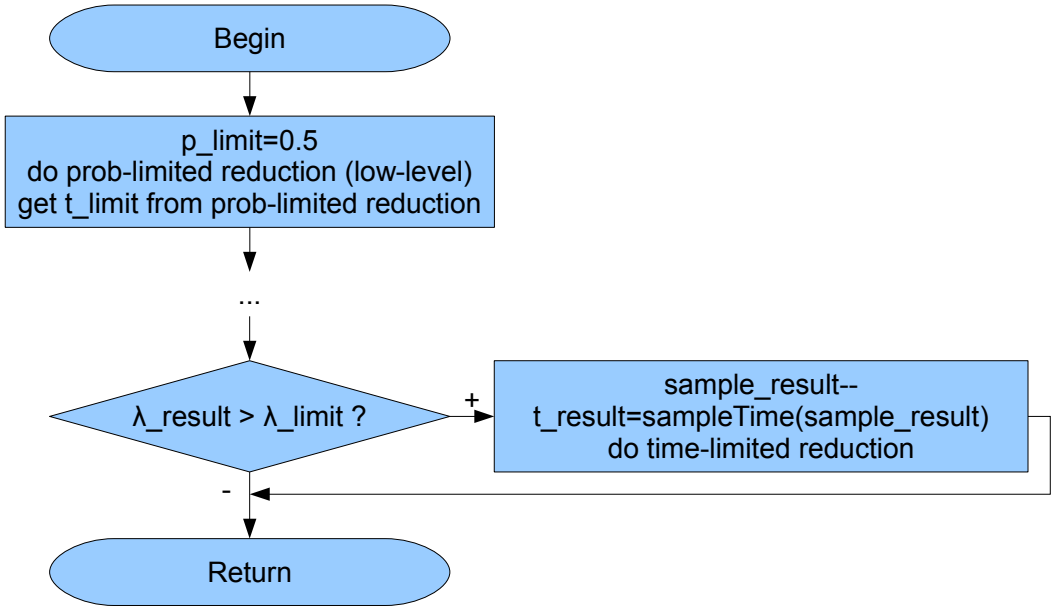

Figure 3.11: Hazard-rate-limited reduction algorithm flowchart (shortened).

Chapter 4

Case Studies and Their Results' Comparisons

The proposed reduction method is demonstrated on two types of case study systems. The first type (see Section 4.1) contains multiple (up to 17) identical dependable blocks configured as an N-modular redundant system (NMR). Models of the internal block redundancy used in the study systems are used as dependability models of railway/subway interlocking equipment used in Czech Republic. The total hazard rate of this system and SIL value is calculated in this case.

NMR-based case study systems are also used to present the partial reduction in Section 4.3 and the ability of the proposed reduction to trade off between accuracy and calculation time in Section 4.4.

The second case study type introduced in Section 4.5 is a model of a modular system with hot-swap repair capability. This case study is used to present the reduction of a three-level heterogeneous dependability model based on Markov chains and reliability block diagrams to calculate MTTF.

4.1 NMR-based Case Studies

Two systems are used as case studies in this section. Both systems use dependable blocks connected in an N-modular Redundant (NMR) system, and both systems are reduced using two-level full reduction.

The dependable blocks used in the first system use Two-out-of-two (2oo2) redundancy [29], [30]. The second system is based on blocks using Modified duplex system (MDS) redundancy [31]. Each dependable block contains two independent copies of functional modules, thus the safety of the blocks using these redundancies cannot be violated by a single fault. The structure of the systems is shown in Fig. 4.1. All blocks used in a system are identical.

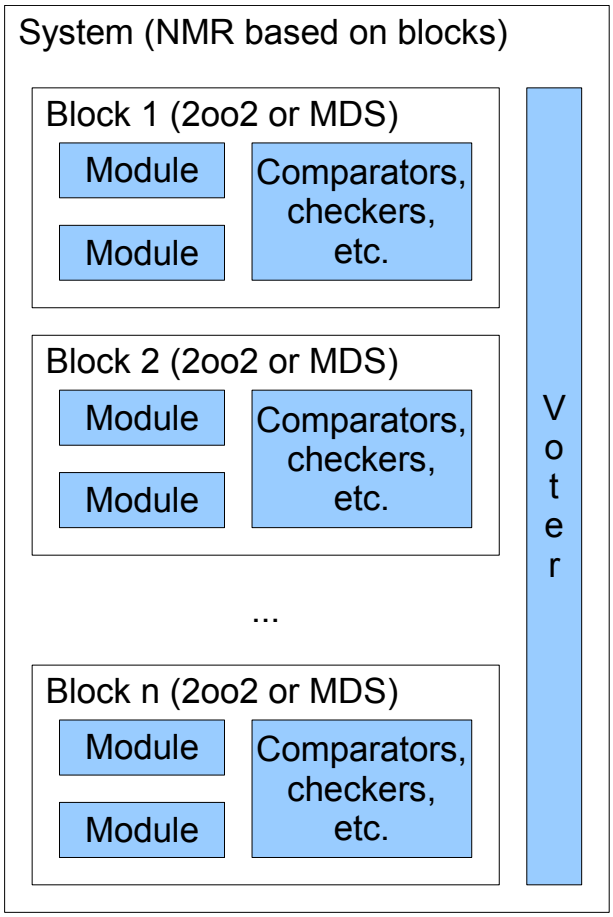

Figure 4.1: Block diagram of case study systems.

4.1.1 Two-out-of-two Block

The Two-out-of-two block is based on two independent modules and a locker (see the basic block diagram in Fig. 4.2). Both modules must be operational to keep the block fully functional. The locker is able to keep the outputs in a safe state – a state, where the block is not functional, but its safety is not violated – and it cannot be affected by a fault. The assumptions and the exact dependability model are taken form [30], where the model is used as a dependability model of the railway station signaling and interlocking equipment.

The dependability model of 2oo2 used in this thesis is constructed under these assumptions:

- Two faults will never occur at the same time.
- Assuming a fault occurs in one module, the locker may lock the block into a safe state using on-line testing techniques (e.g. comparators and/or parity checking – the detailed implementation is not necessary to be known for the calculation). If this block-lock is successful, a possible future fault will not cause a hazard state. A safe

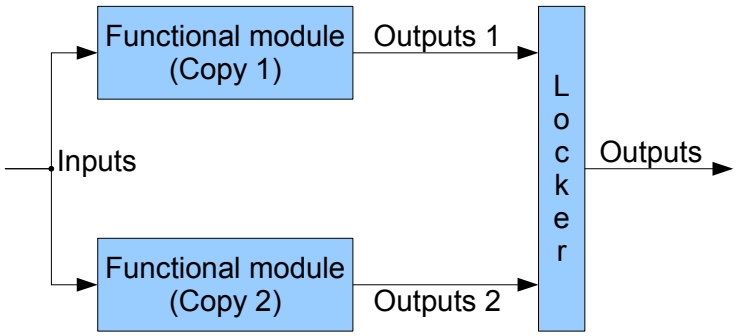

Figure 4.2: Block diagram of the Two-out-of-two block.

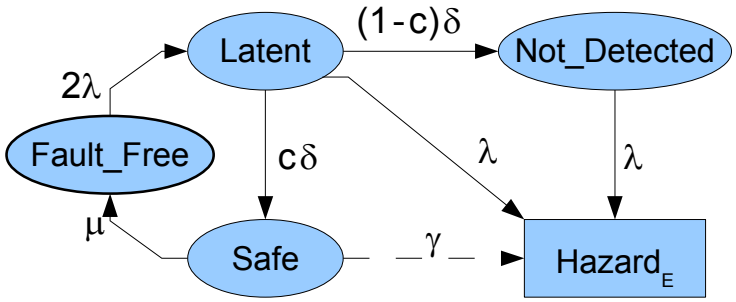

Figure 4.3: Dependability model of Two-out-of-two block used to calculate the exact model failure distribution function.

state is considered as the situation where the block is not operational, but the safety is not violated (all lights are red and the traffic is operated by a human operator in the railway interlocking equipment case).

- If another fault occurs in another (unaffected) module before the block is locked, the safety of the block can be violated. This double-fault situation is considered as a hazard state.

Exact Dependability Model

The model shown in Fig. 4.3 is used to calculate the exact model failure distribution function $F_E(t)$ of the 2oo2 block.

$Fault\_Free$ is the functional/fault-free state of the block. The fault rate of the first fault is 2λ, because the first fault can affect any of the two functional modules of the block.

The *Latent* state is active when the block contains a fault that has not been detected yet. The rate of the on-line test (the inverted average delay between fault origin and detection) is labeled as δ. If the test is performed successfully (a fault is detected), the block will be locked in the $Safe$ state. The probability of a successful test is labeled as c.

If the test fails (the fault is present, but not detected), the block will be in the *Not_Detected* state. The safety of the block is not violated in this state, but another fault (with a fault rate λ) affecting the unaffected functional module will lead to safety violation ($Hazard_E$ state). The second fault hit inside already affected functional module cannot cause a hazard, because the other functional module works correctly.

The arc leading from *Latent* to $Hazard_E$ expresses the probability that a second fault will affect the unaffected functional module before the test is finished.

The block locked in the $Safe$ state waits until the repair is finished (repair rate μ). The block is not functional in this state, but the safety is not violated.

The functionality of the block will be provided by a backup/emergency method (by a human operator in this case), when the block is locked in the $Safe$ state. The rate γ expresses the hazard rate of the backup/emergency method (a mistake of a human operator). This rate should be included into the safety analysis if a more complex analysis needs to be done.

The probability of detection of a fault, the fault rate, and the block-lock rate of the block form the following parameters values. The values have been taken from [30].

$\mu = 24^{-1}\,[h^{-1}]$ – the repair rate

$\lambda = 10^{-5}\,[h^{-1}]$ – the fault rate

$\delta = 10^{-1}\,[h^{-1}]$ – the block-lock rate

$c = 0.6$ – the probability of detecting a fault by the block-lock

$\gamma = 10^{-3}\,[h^{-1}]$ – the backup/emergency method hazard rate

Dependability Model Reduction

The reduced model of the 2oo2 block is the same as shown in the right part of the illustrative example in Fig. 3.1.

The steps of reduction correspond to the algorithm described in Section 3.1. The parameters values are as follows:

$minStep = 0.1\%$

$Samples\ per\ decade = 100$

Calculate the exact model failure distribution function $F_E(t)$.

The following system of differential equations describing the dependability model of 2oo2 block is used for the calculation:

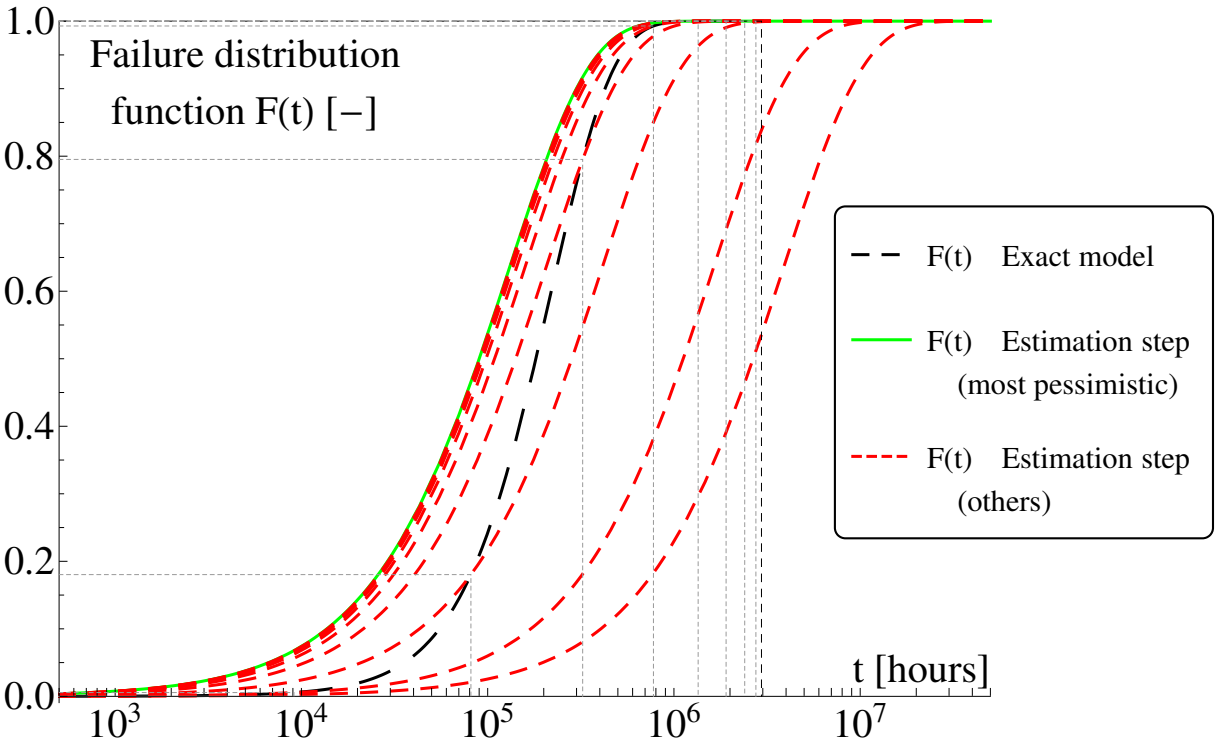

Figure 4.4: Failure distribution function of the exact model ($F_E(t)$) and estimated functions intersecting $F_E(t)$ at predefined time levels (Two-out-of-two block).

$$
\begin{aligned}
p'_{Fault\_Free}(t) &= p_{Safe}(t)\,\mu - p_{Fault\_Free}(t)\,2\lambda \\
p'_{Latent}(t) &= p_{Fault\_Free}(t)\,2\lambda - p_{Latent}(t)\,\delta - p_{Latent}(t)\,\lambda \\
p'_{Not\_Detected}(t) &= p_{Latent}(t)\,(1-c)\,\delta - p_{Not\_Detected}(t)\,\lambda \\
p'_{Safe}(t) &= p_{Latent}(t)\,c\,\delta - p_{Safe}(t)\,(\mu+\gamma) \\
p'_{Hazard_E}(t) &= p_{Safe}(t)\,\gamma + p_{Latent}(t)\,\lambda + p_{Not\_Detected}(t)\,\lambda \\
p_{Fault\_Free}(0) &= 1 \\
p_{Latent}(0) &= p_{Not\_Detected}(0) = p_{Safe}(0) = p_{Hazard_E}(0) = 0
\end{aligned}
$$

Find an estimated value λ_{Hazard_Est}.

All 10 time intersection levels and estimations are shown in Fig. 4.4. The horizontal axis represents the time of operation measured in hours, the vertical axis represents the failure distribution function values. The black long-dashed line represents the exact model failure distribution function, the green line represents the most pessimistic estimated failure distribution function that is used in the correction step, and the red short-dashed lines represent the other estimated failure distribution functions. The horizontal and vertical

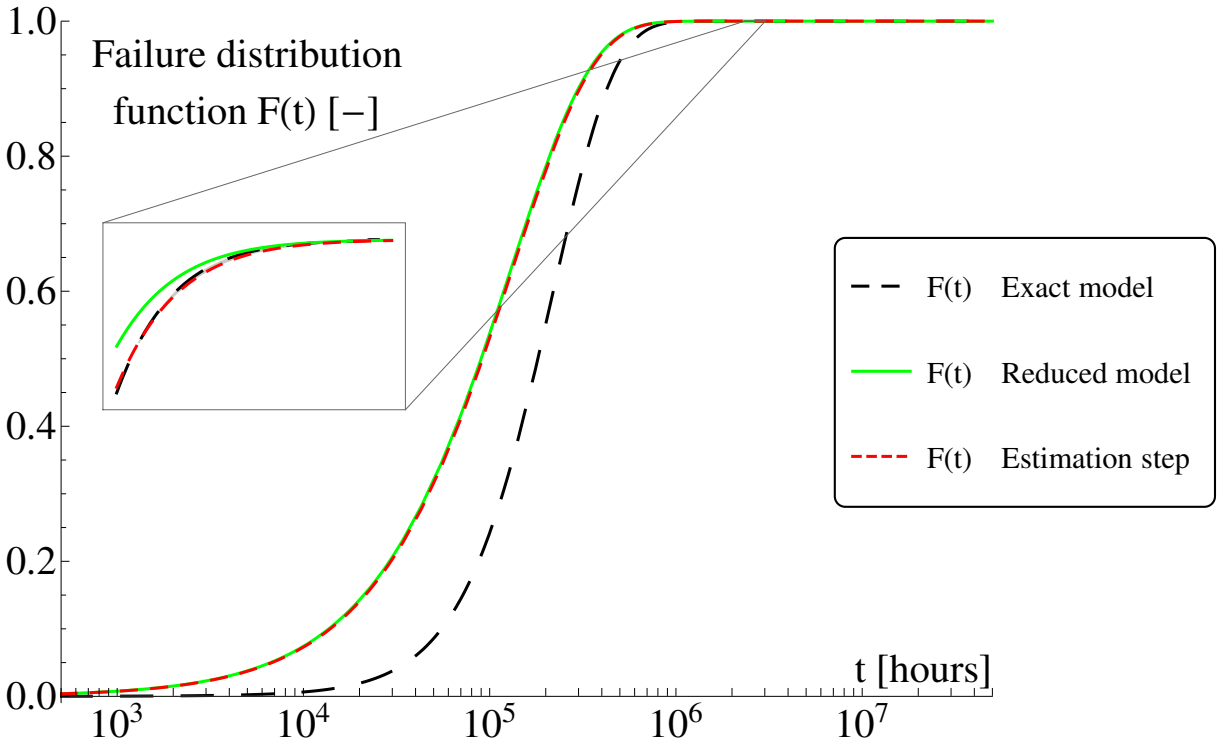

Figure 4.5: Estimated failure distribution functions and failure distribution functions of exact and reduced model of the Two-out-of-two block.

lines show the intersections of the estimated and exact model failure distribution functions. The exact model failure distribution function is greater than the reduced model failure distribution function (i.e. the main requirement is not met) beyond these intersections.

The estimated value λ_{Hazard_Est} taken from the most pessimistic estimated failure distribution function is

$$\lambda_{Hazard\_Est} = 7.694 \times 10^{-6}\,[h^{-1}]$$

Make correction of λ_{Hazard_Est} to satisfy the main requirement.

The estimated failure distribution function from the previous step does not meet the main requirement defined in Section 3.1, because there is an area, where the exact model failure distribution function is greater than the reduced model failure distribution function. The correction made according to the flowchart (see Fig. 3.2 in Section 3.1) is necessary in such case.

The plot shown in Fig. 4.5 shows the exact model failure distribution function, the estimated failure distribution function from the previous step, and the reduced model failure distribution function. The axes are identical to the axes used in the previous plot. The black long-dashed line represents the exact model failure distribution function, the red short-dashed line represents the estimated failure distribution function, and the green line represents the reduced model failure distribution function. The area, where the main

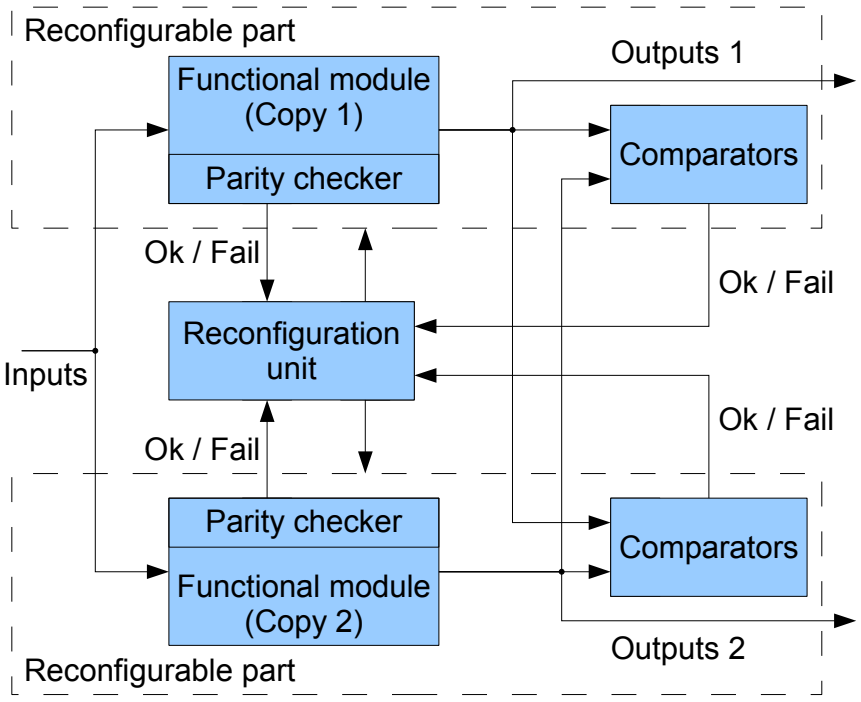

Figure 4.6: Block diagram of the Modified duplex system block.

requirement is not met, is highlighted by a light-gray shading (the area is very small, thus it is hardly observable even in the zoom window).

The corrected value λ_{Hazard_2oo2} is

$$\lambda_{Hazard\_2oo2} = 7.702 \times 10^{-6}\,[h^{-1}]$$

thus the system can be classified as SIL1.

The CPU-time[1] spent on reducing the 2oo2 dependability model is

$$t_{Reduction\_2oo2} = 0.0923\,[s]$$

This CPU-time will be used in Section 4.2.2 to compare runtimes of reduction of hierarchical and exact dependability models of the system using 2oo2 as a block.

4.1.2 Modified Duplex System Block

The Modified Duplex System is based on two independent modules with parity checkers attached [31]. The parity checkers are able to detect some faults. The remaining faults are detected by comparators attached to the outputs of both modules (see the block diagram in Fig. 4.6).

[1]Running on Intel Core i5 @3.3 GHz, OS: Win7 64-bit

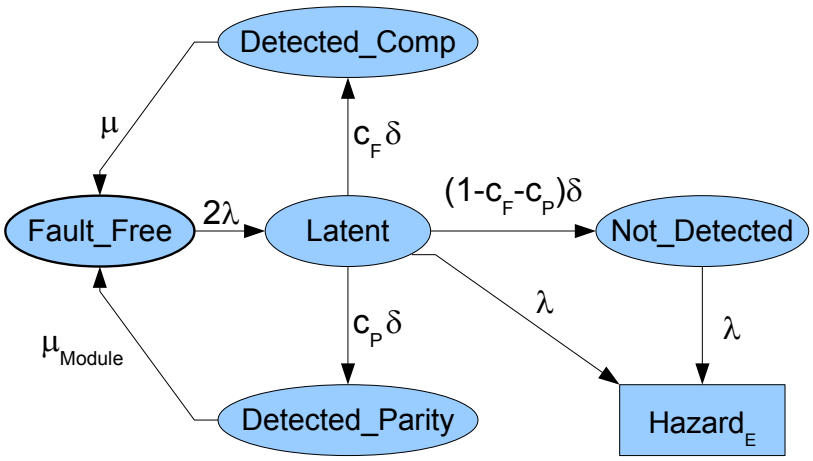

Figure 4.7: Dependability model of the Modified duplex system block used to calculate the exact model failure distribution function.

The MDS is designed to utilize the reconfiguration ability of an FPGA. FPGA is an integrated circuit designed to be configured by a customer or a designer after manufacturing. A part of the FPGA affected by a fault can be repaired by reconfiguration in tenths or hundreds of milliseconds.

The dependability model of MDS used in this thesis is constructed under these assumptions:

- Two faults will never occur at the same time.
- When a fault occurs in one module, the parity checker attached to this module can detect the fault. The parity checker needs not cover all possible faults. If the fault is detected by the parity checker, the affected module is repaired. If the fault is not detected by the parity checker, it can be detected by comparators. Both modules and comparators have to be repaired in such case, because the faulty module cannot be identified. When a fault occurs in the comparator, both modules and comparators have to be repaired.
- If another fault occurs before the repair is completed, the safety of the block can be violated. This double-fault situation is considered as a hazard state.

Exact Dependability Model

The model shown in Fig. 4.7 is used to calculate the exact model failure distribution function $F_E(t)$ of the MDS block.

The states of the model are similar to the states of the model of 2oo2 block. $Fault\_Free$ is the functional/fault-free state of the block. The fault rate of the first fault is 2λ, because the first fault can affect any of the two functional modules of the block. The block is in the *Latent* state when it contains a fault that has not been detected yet.

The fault detection rate is labeled as δ. If a fault is detected by the parity checkers, the block will be locked in the *Detected_Parity* state. The probability of detecting a fault by parity checkers is labeled as c_P. If a fault is detected by comparators, the block will be locked in the *Detected_Comp* state. The probability of detecting a fault by comparators only is labeled as c_F.

If the fault is detected neither by parity checkers nor by comparators, the block will be in *Not_Detected* state. The safety of the block is not violated in this state, but another fault (with fault rate λ) affecting the unaffected functional module will lead to safety violation ($Hazard_E$ state). The second fault hit inside an already affected functional module cannot cause a hazard, because the other functional module works correctly.

The arc leading from *Latent* to $Hazard_E$ expresses the probability that a second fault affects the unaffected functional module before the first fault is detected.

The block locked in the *Detected_Parity* state waits until the repair is finished (repair rate μ_{Module} – only one module is repaired). The block locked in the *Detected_Comp* state also waits until the repair is finished (repair rate μ – both modules and both comparators are repaired). The block is not functional in these states, but the safety is not violated.

The probability of detection of a fault, the fault rate, and the self-test rate of the block form the following parameters values. The values are similar to the 2oo2 block model, but the repair rates are much higher (because of fast reconfiguration).

$\mu = 10^3\,[h^{-1}]$ – the repair rate of the whole block (both modules and both comparators)

$\mu_{Module} = 5 \times 10^3\,[h^{-1}]$ – the repair rate of the faulty module

$\lambda = 10^{-5}\,[h^{-1}]$ – the fault rate

$\delta = 10^{-1}\,[h^{-1}]$ – the fault detection rate

$c_P = 0.6$ – the probability of detecting a fault by the parity checkers

$c_F = 0.2$ – the probability of detecting a fault by the comparators

Dependability Model Reduction

The reduced model of the MDS block is the same as shown in the right part of the illustrative example in Fig. 3.1.

The steps of reduction correspond to the algorithm described in Section 3.1. The parameters values are the same as in the case of the 2oo2 block model:

$minStep = 0.1\%$

$Samples\ per\ decade = 100$

Calculate the exact model failure distribution function $F_E(t)$

The system of differential equations describing the dependability model of MDS block is used for the calculation:

$$
\begin{aligned}
p'_{Fault\_Free}(t) &= p_{Detected\_Comp}(t)\,\mu + p_{Detected\_Parity}(t)\,\mu_{Module} - p_{Fault\_Free}(t)\,2\lambda \\
p'_{Latent}(t) &= p_{Fault\_Free}(t)\,2\lambda - p_{Latent}(t)\,\delta - p_{Latent}(t)\,\lambda \\
p'_{Not\_Detect}(t) &= p_{Latent}(t)\,(1 - c_F - c_P)\,\delta - p_{Not\_Detect}(t)\,\lambda \\
p'_{Detected\_Comp}(t) &= p_{Latent}(t)\,c_F\,\delta - p_{Detected\_Comp}(t)\,\mu \\
p'_{Detected\_Parity}(t) &= p_{Latent}(t)\,c_P\,\delta - p_{Detected\_Parity}(t)\,\mu_{Module} \\
p'_{Hazard_E}(t) &= p_{Latent}(t)\,\lambda + p_{Not\_Detect}(t)\,\lambda \\
p_{Fault\_Free}(0) &= 1 \\
p_{Not\_Detect}(0) &= p_{Detected\_Comp}(0) = p_{Hazard_E}(0) = p_{Detected\_Parity}(0) = p_{Latent}(0) = 0
\end{aligned}
$$

Find an estimated value λ_{Hazard_Est} and make correction

We use the same method as presented in Section 4.1.1.

The estimated value λ_{Hazard_Est} taken from the most pessimistic estimated failure distribution function is

$$\lambda_{Hazard\_Est} = 3.912 \times 10^{-6}\,[h^{-1}]$$

The corrected value λ_{Hazard_MDS} is

$$\lambda_{Hazard\_MDS} = 3.916 \times 10^{-6}\,[h^{-1}]$$

thus the system can be classified as SIL1.

The CPU-time[2] spent on reducing the MDS dependability model is

$$t_{Reduction\_MDS} = 0.0994\,[s]$$

This CPU-time will be used in Section 4.2.2 to compare runtimes of reduction of hierarchical and exact dependability models of the system using MDS as a block.

4.1.3 N-modular Redundancy

N-modular Redundancy (NMR) is based on N identical blocks and a voter. This voter is able to compare all outputs of the blocks. It uses majority voting to produce a single output. If less than half of the blocks fail, the voter is able to produce correct output. If more than half of the blocks fail, the voter will produce an incorrect output – this situation is considered as a hazard state. The erroneous blocks cannot be identified, thus there is no restoration/repair possibility.

Exact Dependability Model

The model shown in Fig. 4.8 is used to calculate the exact model failure distribution function of a generic NMR system. The NMR system containing N blocks will contain

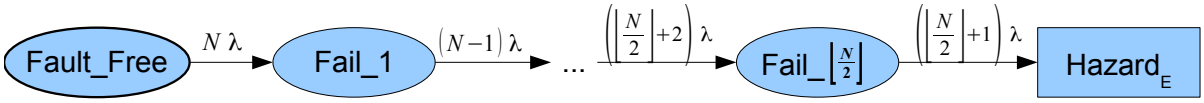

Figure 4.8: Dependability model of generic N-modular redundant system used to calculate exact model failure distribution function.

$\lfloor \frac{N}{2} \rfloor$ transient states. These states correspond to the blocks that are in the hazard state. NMR systems consisting of 3 to 17 blocks are reduced in this thesis.

Dependability Model Reduction

The reduced model of the NMR block is the same as shown in the right part of the illustrative example in Fig. 3.1.

The steps of reduction correspond to the algorithm described in Section 3.1. The reduction parameters values are the same as in the cases of 2oo2 and MDS blocks models:

$$minStep = 0.1\%$$

$$Samples\ per\ decade = 100$$

Calculate the exact model failure distribution function $F_E(t)$

The system of differential equations describing the dependability model of NMR containing N blocks is used for the calculation:

$$\begin{aligned}
p'_{Fault_Free}(t) &= -p_{Fault_Free}(t)\, N\, \lambda \\
p'_{Fail_1}(t) &= p_{Fault_Free}(t)\, N\, \lambda - p_{Fail_1}(t)\, (N-1)\, \lambda \\
p'_{Fail_2}(t) &= p_{Fail_1}(t)\, (N-1)\, \lambda - p_{Fail_2}(t)\, (N-2)\, \lambda \\
&\dots \\
p'_{Fail_\lfloor \frac{N}{2} \rfloor}(t) &= p_{Fail_\lfloor \frac{N}{2} \rfloor - 1}(t) \left(\left\lfloor \frac{N}{2} \right\rfloor + 2 \right) \lambda - p_{Fail_\lfloor \frac{N}{2} \rfloor}(t) \left(\left\lfloor \frac{N}{2} \right\rfloor + 1 \right) \lambda \\
p'_{Hazard_E}(t) &= p_{Fail_\lfloor \frac{N}{2} \rfloor}(t) \left(\left\lfloor \frac{N}{2} \right\rfloor + 1 \right) \lambda \\
p_{Fault_Free}(0) &= 1 \\
p_{Fail_1}(0) &= p_{Fail_2}(0) = \dots = p_{Fail_\lfloor \frac{N}{2} \rfloor}(0) = 0 \\
p_{Hazard_E}(0) &= 0
\end{aligned}$$

Find an estimated value λ_{Hazard_Est} and make correction

The method presented in Section 4.1.1 is used.

The hazard rate of the NMR model depends on the hazard rate of the block. The results of the NMR based on 2oo2 and MDS blocks are shown in the following section.

[2]Running on Intel Core i5 @3.3 GHz, OS: Win7 64-bit

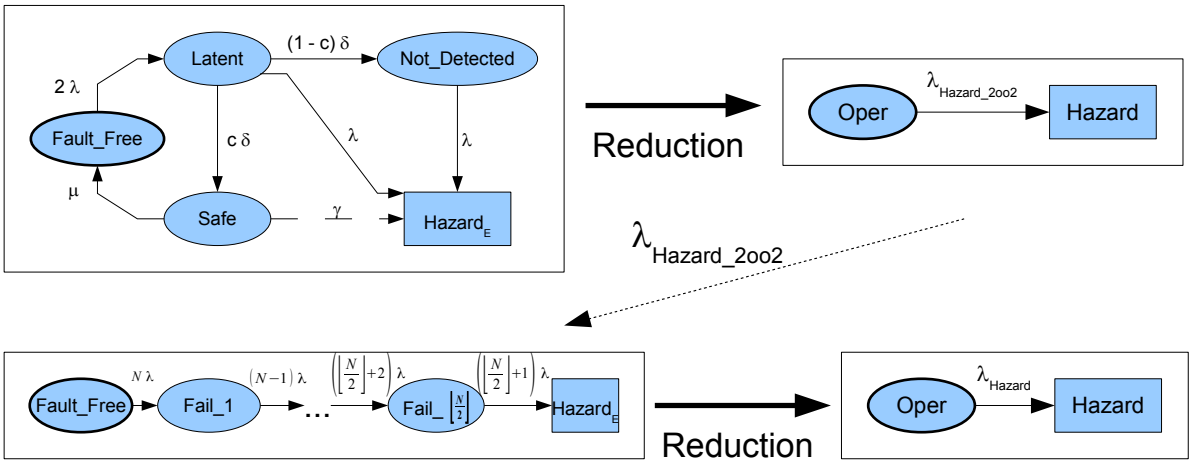

Figure 4.9: Hierarchical dependability model of case study system (NMR based on 2oo2 blocks).

4.2 Hierarchical Models

The case study systems used in this thesis originate from railway interlocking systems, thus the primary results of concern are the hazard rates (and SILs), but the calculation time is also important, especially when a large system containing many blocks is calculated. Improper method can lead to unacceptable calculation time or inadequate accuracy.

4.2.1 NMR based on Two-out-of-two or Modified Duplex System Blocks

The hierarchical dependability model of the NMR system based on 2oo2 blocks is shown in Fig. 4.9. The hierarchical dependability model of the NMR system based on MDS blocks is similar – the model of a 2oo2 block is replaced by the model of a MDS block only.

The model of a 2oo2/MDS block is created, reduced, and the result of reduction (λ_{Hazard_2oo2} calculated in Section 4.1.1 or λ_{Hazard_MDS} calculated in Section 4.1.2 respectively) is taken as the hazard rate (λ) of the NMR model. The results of the reduction of the hierarchical dependability model (λ_{Hazard}) are calculated in Section 4.2.

The exact model of the NMR based on 2oo2 blocks (see Fig. 4.10) is the result of Cartesian product of N models of the blocks and the model of the NMR. The exact model of the NMR based on MDS blocks is similar – the model of a 2oo2 block is replaced by the model of a MDS block once again.

4.2.2 Comparison of Runtimes

Table 4.1 shows the comparison of CPU-times of solutions of an N-modular redundant system based on identical Two-out-of-two or Modified duplex system blocks. The first

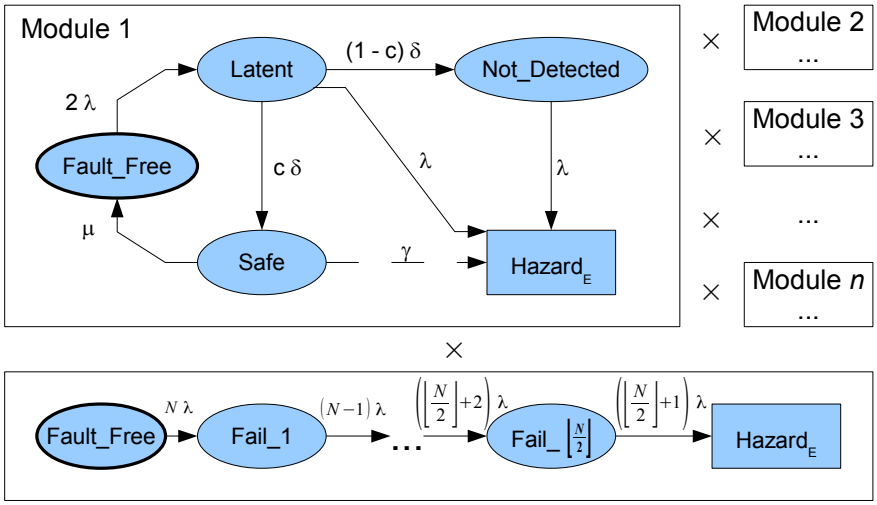

Figure 4.10: The models used to create exact dependability model of the case study system (NMR based on 2oo2 blocks).

Table 4.1: Number of states and CPU-times of solutions of N-modular redundant system based on identical blocks.

NMR blocks	Two-out-of-two blocks			Modified duplex system blocks		
	Exact model states	NDSolve time [s]	Reduction time [s]	Exact model states	NDSolve time [s]	Reduction time [s]
n01 [1)]	5	0.0127	0.0923 [2)]	6	0.0173	0.0994 [2)]
n03	34	0.0414	0.175	55	0.0549	0.185
n05	121	0.123	0.174	246	0.263	0.183
n07	315	0.315	0.175	771	0.770	0.185
n09	680	0.707	0.176	1,946	2.997	0.185
n11	1,295	1.721	0.176	4,242	10.58	0.184
n13	2,254	4.089	0.177	8,316	29.25	0.185
n15	3,666	7.139	0.176	15,042	117.9	0.185
n17	5,655	12.10	0.177	25,542	477.5	0.187
			...			
n99	4,150,550	10^9 yrs[3)]	0.198	89,092,835	10^{18} yrs[3)]	0.208

[1)] NMR containing one block is equivalent to a single 2oo2/MDS block.
[2)] Time to reduce a 2oo2/MDS block only.
[3)] Estimation based on the exponential extrapolation of systems n3–n17.

column shows the number of the blocks, the second column shows the number of states of the exact dependability model.

The next two columns show the CPU-times spent on solving[3] the systems of differential equations of the dependability models. The CPU-time spent on solving the system of the exact dependability model – the model generated by the Cartesian product of the dependability models of the 2oo2 blocks configured as NMR – is shown in the third column. The fourth column shows the CPU-times spent on reducing the 2oo2 dependability model ($0.0923\,s$ – $t_{Reduction\_2oo2}$ taken from Section 4.1.1) and CPU-time spent on reducing the dependability model of NMR. The reduction time includes the time required to solve the exact model, to sample the exact failure distribution function, to estimate λ_{Hazard_Est}, and to make the correction. Both models (2oo2 and NMR) are small, thus the most of the time is spent on the corrections.

The second three columns show the same results for the case study system based on Modified duplex system blocks.

The CPU-time spent on solving the Cartesian model grows rapidly with increasing number of the blocks, but the reduction time is nearly constant[4]. Therefore the reduction will be faster when a system containing more than 7 blocks is used in these particular configurations.

The plot in Fig. 4.11 shows the exponential progress of the CPU-time spent on solving the system of the exact dependability model (NDSolve time) with respect to the number of the MDS blocks. The horizontal axis of the plot represents the number of the MDS blocks, the vertical axis (plotted on a logarithmic scale) represents the NDSolve time. The points in the plot are NDSolve times of the n3–n17 systems, the dashed line is an exponential interpolation.

This interpolation is used to estimate NDSolve time of the system containing 99 blocks (n99) in the last row of Table 4.1. The reduction time of the system n99 remains nearly constant.

4.2.3 Hierarchy Reduction Error

Table 4.2 shows the difference between the hazard rates calculated using the hierarchical model and the hazard rate calculated by reducing the Cartesian model directly. The first column shows the number of the 2oo2/MDS blocks, the second column shows the hazard rate of the NMR system based on 2oo2 blocks using the reduction of the Cartesian model. The third column shows the hazard rate of the NMR system using a two-level reduction of the hierarchy model. The fourth column contains the ratio of the hazard rates shown in the previous columns.

[3]NDSolve command of Mathematica 9.01 [32] software running on Intel Core i5 @3.3 GHz, OS: Win7 64-bit

[4]The measured CPU-times varies by ca. ±10%, when the calculation is performed repeatedly, due to inaccurate time values provided by Mathematica. This variance remains intact, when 100 repetitions of the same instance are calculated.

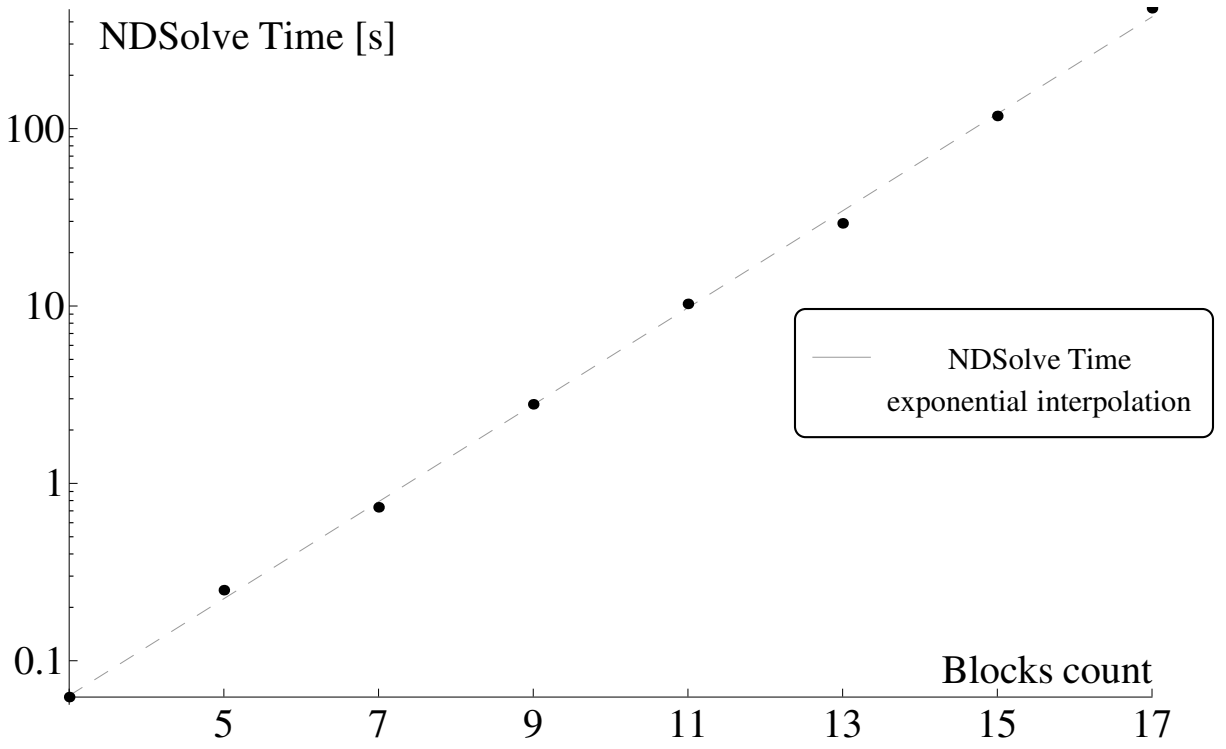

Figure 4.11: CPU-time spent on solving the system of the exact dependability model with respect to the the number of the Modified duplex system blocks.

Table 4.2: Comparison of hazard rates of N-modular redundant system calculated using hierarchy and Cartesian-product safety models

	Two-out-of-two blocks			**Modified duplex system blocks**		
NMR blocks	$\lambda_{\mathbf{Hazard\_Cart}}$ $[\times\mathbf{10^{-6}\,h^{-1}}]$	$\lambda_{\mathbf{Hazard\_Hier}}$ $[\times\mathbf{10^{-6}\,h^{-1}}]$	$\frac{\lambda_{\mathbf{Hazard\_Hier}}}{\lambda_{\mathbf{Hazard\_Cart}}}$	$\lambda_{\mathbf{Hazard\_Cart}}$ $[\times\mathbf{10^{-6}\,h^{-1}}]$	$\lambda_{\mathbf{Hazard\_Hier}}$ $[\times\mathbf{10^{-6}\,h^{-1}}]$	$\frac{\lambda_{\mathbf{Hazard\_Hier}}}{\lambda_{\mathbf{Hazard\_Cart}}}$
n03	13.88	14.71	1.060	7.322	7.476	1.021
n05	18.89	20.98	1.111	10.30	10.68	1.037
n07	23.10	26.66	1.154	12.86	13.54	1.053
n09	26.81	31.73	1.184	15.12	16.18	1.070
n11	29.94	36.36	1.214	17.06	18.56	1.088
n13	32.69	40.83	1.249	18.88	20.71	1.097
n15	35.31	44.79	1.268	20.35	22.71	1.116
n17	37.37	48.20	1.290	21.82	24.65	1.130

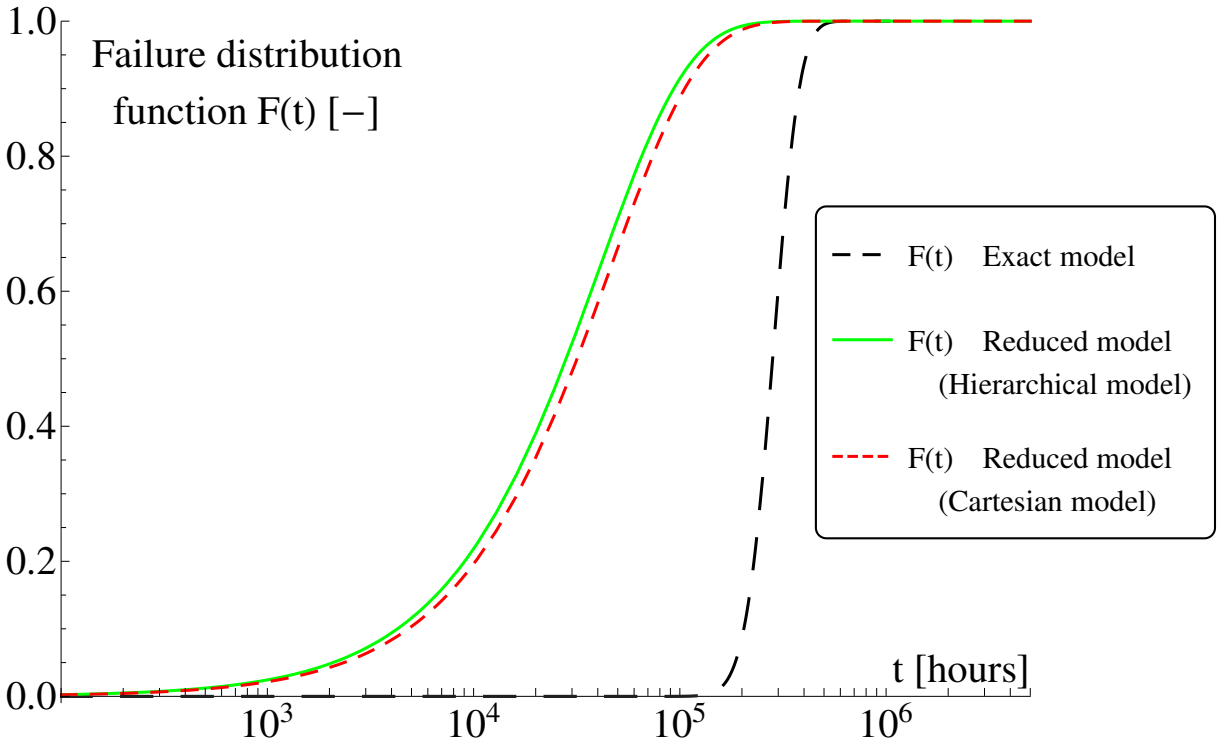

Figure 4.12: Comparison of failure distribution functions of N-modular redundant system based on 17 identical Modified duplex system blocks.

The second three columns show the same results for the case study system based on Modified duplex system blocks.

The direct reduction of the Cartesian model leads to the most accurate results, but it can become computationally impossible in practice (see Table 4.1 – NDSolve time column).

The plot in Fig. 4.12 shows the comparison of the failure distribution functions of the N-modular redundant system based on 17 identical MDS blocks. The horizontal axis of the plot represents the time of operation measured in hours, the vertical axis represents the failure distribution function values. The black long-dashed line represents the exact model failure distribution function, the red short-dashed line represents the reduced model failure distribution function calculated using the Cartesian model, and the green line represents the reduced model failure distribution function calculated using the hierarchy model.

The reduction of this system (NMR17) is the most inaccurate one of the systems based on MDS blocks calculated in this thesis. The difference between hazard rates of the Cartesian and hierarchical model is ca. 13% (the difference in 2oo2-NMR case is ca. 29%), but both failure distribution functions of the reduced models differ significantly from the exact model function (especially in the lower part of the plot, where the standard usage of the modeled system is expected). The function of the Cartesian and hierarchical models reaches 10% probability of failure in ca. 4,300 and 4,800 hours respectively, but the exact model function reaches the same probability in ca. 209,000 hours. This difference is

unacceptable – the full reduction is unsuitable for this kind of systems, thus the partial reduction should be used to obtain appropriate results (see the partial reduction of the system based on MDS blocks in Section 4.3).

4.3 Partial Reduction

This section contains the results of the partial reduction of the system MDS-NMR17, as the most inaccurate representative of MDS-NMR systems in Section 4.2.3. The three different possibilities of partial reduction described in Section 3.2 are used to reduce both levels of hierarchy of the model:

- *Time-limited reduction* – uses t_{limit} itself
- *Probability-limited reduction* – uses a p_{limit} probability
- *Hazard-rate-limited reduction* – uses a λ_{limit} hazard rate

4.3.1 Time-limited Reduction

Table 4.3 shows the progression of the hazard rates (the second column) and SILs [6] (the third column) of the hierarchical model depending on the t_{limit} time (the first column) value used in the partial reduction. The table also shows the probability of failure according to the failure distribution function of the reduced model (F_R) at the time t_{limit} (the fourth column labeled as $F_R(t_{limit})$). The last two columns show the times when the failure distribution functions of different models (reduced model using full reduction in the fifth column, and the exact model function in the last column) reaches the *limit* probability value $F_R(t_{limit})$ shown in the fourth column.

The results obtained using the limit value (200,000 $hours$) will be compared to the other methods of the partial reduction (probability-limited using $p_{limit} = 0.1$ and hazard-rate-limited using $\lambda_{limit} = 0.5 \times 10^{-6}\, h^{-1}$) in Section 4.3.4.

The hazard rate of the model using the partial reduction is significantly decreased. Using $t_{limit} = 200{,}000\, hours$ during the model reduction decreases hazard rate ca. 50 times. That means that the system not meeting the requirements to be classified as SIL1 can be classified as SIL2 when the partial reduction is used. The preventive maintenance period of the system has to be lower than 200,000 hours (ca. 22 years), the main requirement and the pessimism of the solution are not guaranteed otherwise.

The plot in Fig. 4.13 shows all the failure distribution functions used in Table 4.3 when $t_{limit} = 200{,}000\, hours$. The horizontal axis of the plot represents the time of operation measured in hours, the vertical axis represents the failure distribution function values. The black long-dashed line represents the exact model failure distribution function, the red short-dashed line represents the reduced model failure distribution function calculated using the full reduction and the green line represents the reduced model failure distribution function calculated using the partial reduction. The vertical line represents the reduction limit t_{limit}.

Table 4.3: Progress of hazard rates and SILs and comparison of runtimes when the failure distribution functions of different models reaches the t_{limit} time value using the time-limited partial reduction.

$\mathbf{t_{limit}}$ [hours]	$\lambda_{\mathbf{Hazard\_Part}}$ [$\times 10^{-6}\,\mathbf{h^{-1}}$]	SIL [-]	$\mathbf{F_R(t_{limit})}$ [-]	$\mathbf{t_{Limit\_Full}}$ [hours]	$\mathbf{t_{Limit\_Exact}}$ [hours]
100,000	861.33×10^{-6}	>4	86.129×10^{-6}	3.4942	102,329
150,000	0.05768	3	0.008615	351.01	151,356
200,000[1)]	**0.5173**	**2**	**0.09828**	**4,197**	**204,174**
250,000	1.5376	1	0.3191	15,594	257,040
300,000	3.1677	1	0.6134	38,552	301,995
350,000	5.1325	1	0.8341	72,875	354,813
400,000	7.1703	1	0.9432	116,352	407,380
450,000	9.0180	1	0.9827	164,627	457,088
500,000	10.561	<1	0.9949	214,216	512,861
550,000	12.526	<1	0.9990	279,474	562,341
None (Full)[2)]	24.650	<1	$\rightarrow 1$	_[3)]	_[3)]

[1)] The plot comparing the failure distribution functions of the exact model, and the reduced model using the full reduction and the partial reduction using $t_{limit} = 200{,}000\,hours$ is shown in Fig. 4.13.
[2)] The partial reduction without reduction limit is equal to the full reduction.
[3)] The failure distribution functions cannot be equal to 1.

4.3.2 Probability-limited Reduction

Table 4.4 shows the progression of the hazard rates (the second column) and SILs (the third column) of the hierarchical model depending on the p_{limit} probability value (the first column) used in the partial reduction. The last two columns show the times when the failure distribution functions of the reduced models (using partial reduction in the fourth column, and using the full reduction in the last column) reaches the p_{limit} probability value.

The results obtained using the limit value (0.1) will be compared to the other methods of the partial reduction (time-limited using $t_{limit} = 200{,}000\,hours$ and hazard-rate-limited using $\lambda_{limit} = 0.5 \times 10^{-6}\,h^{-1}$) in Section 4.3.4.

The plot in Fig. 4.14 shows all the failure distribution functions used in Table 4.4 when $p_{limit} = 0.1$. The meaning of the axes and the lines is similar to the plot shown in Fig. 4.13.

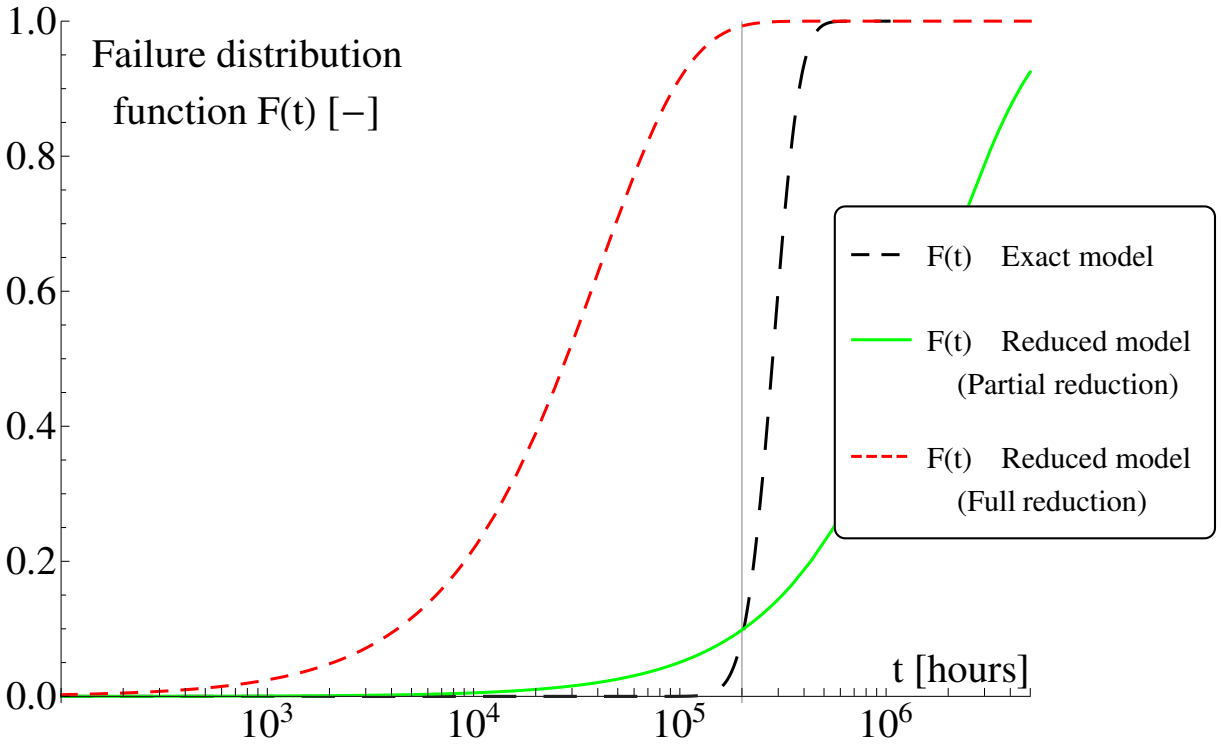

Figure 4.13: Comparison of failure distribution functions of exact and reduced model using full and partial reduction using reduction limit $t_{limit} = 200{,}000\,hours$.

4.3.3 Hazard-rate-limited Reduction

Table 4.5 shows the progression of the probability of failure (the third column labeled as $F_R(t_{limit})$), when the failure distribution function of the reduced model meets the failure distribution function of the exact model, depending on the λ_{limit}/SIL value (the first and the second column) used in the partial reduction. The last two columns show the difference between the times when the failure distribution functions of the reduced models (using partial reduction in the third column, and using the full reduction in the last column) reaches the probability of failure shown in the third column.

The results obtained using the limit value ($0.5 \times 10^{-6}\,h^{-1}$) will be compared to the other methods of the partial reduction (time-limited using $t_{limit} = 200{,}000\,hours$ and probability-limited using $p_{limit} = 0.1$) in Section 4.3.4.

4.3.4 Comparison of Partial Reduction Types

Table 4.6 shows that all three types of the partial reduction leads to the similar results, when the limits leading to the similar values are used. Please note that the values are not identical. If the limit values leading to the identical values would be used, the other values would be identical as well (e.g. the time-limited reduction using $t_{limit} = 213{,}796\,hours$

Table 4.4: Progress of hazard rates and SILs and comparison of runtimes when the failure distribution functions of different models reaches the p_{limit} probability value using the probability-limited partial reduction.

$\mathbf{p_{limit}}$ [−]	$\lambda_{\mathbf{Hazard\_Part}}$ [$\times\mathbf{10^{-6}\,h^{-1}}$]	**SIL** [-]	$\mathbf{t_{limit}}$ [**hours**]	$\mathbf{t_{Limit\_Full}}$ [**hours**]
0.0001	0.001115	4	104,713	4.6268
0.0010	0.008216	4	125,893	41.006
0.0100	0.06973	3	158,489	438.09
0.1000[1)]	**0.5173**	**2**	**208,930**	**4,284**
0.3500	1.7024	1	263,027	17,752
0.6000	3.1677	1	309,030	38,808
0.9500	7.5302	1	426,580	127,347
0.9900	9.7844	1	489,779	189,981
0.9990	12.526	<1	575,440	285,745
1 (Full)[2)]	24.650	<1	–[3)]	–[3)]

[1)] The plot comparing the failure distribution functions of the exact model, and the reduced model using the full reduction and the partial reduction using $p_{limit} = 0.1$ is shown in Fig. 4.14.

[2)] The partial reduction with reduction limit 1 is equal to the full reduction.

[3)] The failure distribution functions cannot be equal to 1.

would lead to failure probability $F_R(213{,}796) = 0.1160$ and the hazard rate $\lambda_{Hazard} = 0.5 \times 10^{-6}\,h^{-1}$).

4.4 Reduction Parameters Impact

Two parameters that determine the trade-off between reduction accuracy and reduction time were introduced in Section 3.1:

1. *minStep* – this parameter determines the end of the bisection loop of the correction.
2. *Samples per decade* – this parameter determines the number of samples that will be tested for the main requirement fulfillment.

The impacts of these parameters on the full reduction of the system MDS-NMR17 are presented in this section. Both models (low- and top-level) are reduced using the full reduction with varying parameters.

Table 4.5: Progress of SILs and comparison of runtimes depending on the selected λ_{limit} value using the hazard-rate-limited partial reduction.

$\lambda_{\mathbf{limit}}$ [$\times\mathbf{10^{-6}\,h^{-1}}$]	**SIL** [-]	$\mathbf{F_R(t_{limit})}$ [-]	$\mathbf{t_{Limit}}$ [hours]	$\mathbf{t_{Limit\_Full}}$ [hours]
0.01	3	0.00101	125,893	41.006
0.02	3	0.00214	134,896	86.698
0.05	3	0.00692	151,356	281.69
0.1	2	0.0133	162,181	543.30
0.2	2	0.0297	177,828	1,223
0.5	**2**	**0.0857**	**204,174**	**3,633**
1	1	0.2013	234,423	9,117
2	1	0.3899	269,153	20,042
5	1	0.8122	354,813	67,834
10	<1	0.9907	489,779	189,981
20	<1	$\rightarrow 1$.[1)]	912,011	722,242

[1)] The probability is close, but not equal to 1, thus the failure distribution functions can be equal to this value.

Table 4.6: Comparison of the selected values of all three types of partial reduction.

Limit type	**Time** [hours]	**Probability** [-]	**Hazard rate** [$\times\mathbf{10^{-6}\,h^{-1}}$]	**SIL** [-]
Time	**200,000**	0.0983	0.5173	2
Probability	208,930	**0.1000**	0.5173	2
Hazard rate	204,174	0.0857	**0.5**	2

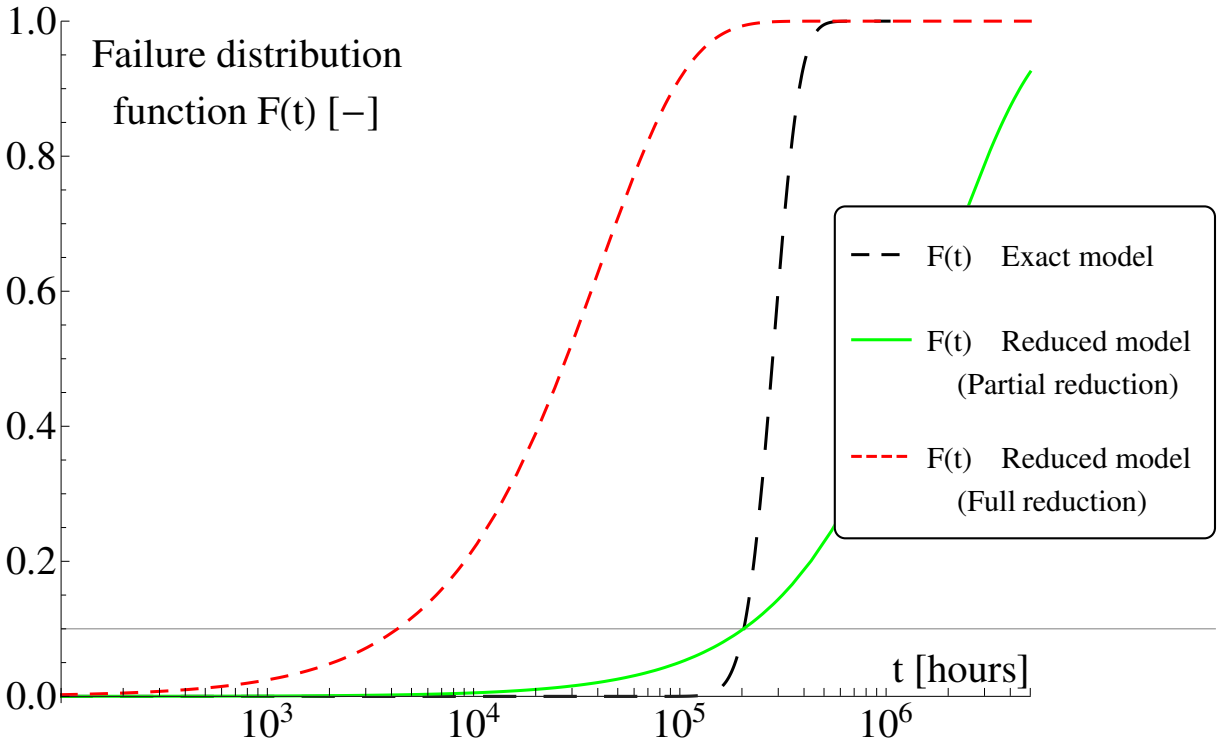

Figure 4.14: Comparison of failure distribution functions of exact model, and reduced model using full and partial reduction using reduction limit $p_{limit} = 0.1$.

4.4.1 minStep

Table 4.7 shows the impact of the *minStep* parameter. The size of *minStep* expressed as a percentage of the estimated value λ_{Hazard_Est} is shown in the first column, the hazard rate of the reduced model and the time of the reduction using selected *minStep* in the second and the third column respectively. The number of *Samples per decade* is 100 during this measurement.

The lower is the *minStep* value, the lower (more precise) is the calculated hazard rate. This dependency is not monotonic, but the error of each level of the reduction is lower than the *minStep*. The calculated value is more pessimistic than the optimal value. The reduction time is slightly increasing[5] when the *minStep* value decreases.

The plot shown in Fig. 3.5 shows the exact model failure distribution function, the estimated failure distribution function from the previous step, and the reduced model failure distribution function. The axes are identical to the axes used in the previous plot. The black long-dashed line represents the exact model failure distribution function, the red short-dashed line represents the estimated failure distribution function, and the green

[5]The measured CPU-times varies by ca. ±10%, when the calculation is performed repeatedly, due to inaccurate time values provided by Mathematica.

Table 4.7: Comparison of hazard rates and reduction times with respect to the accuracy of the correction step.

minStep size [%]	$\lambda_{\mathbf{Hazard}}$ $[\times \mathbf{10^{-6}\,h^{-1}}]$	**Reduction time [s]**
10	29.517	0.1863
5	26.889	0.1873
2	25.552	0.1868
1	24.953	0.1885
0.5	24.656	0.1860
0.2	24.709	0.1869
0.1[1)]	**24.650**	**0.1863**
0.05	24.621	0.1868
0.02	24.603	0.1865
10^{-8}	24.591	0.5994

[1)] Default value used in this thesis.

line represents the reduced model failure distribution function. The area, where the main requirement is not met, is highlighted by a light-gray shading (see the zoom window).

The plot in Fig. 4.15 shows the boundary of the failure distribution function when the worst, the default, and the best *minStep* values shown in Table 4.7 are used. The horizontal axis of the plot represents the time of operation measured in hours, the vertical axis represents the failure distribution function values. The black long-dashed line represents the function using the best value, the green line represents the failure distribution function using default settings, the red short-dashed line represents the worst possible failure distribution function. The hazard rate of the worst possible function is $29.517 \times 10^{-6}\,h^{-1}$ that is ca. 20% higher than the hazard rate calculated using the best value. The 20% difference is caused by the hierarchical approach – the maximal 10% difference is applied at each level of the hierarchy. The hazard rate of the function using the default value is $24.650 \times 10^{-6}\,h^{-1}$ – ca. 0.24% higher than the hazard rate using the best value – but it is calculated 3 times faster.

4.4.2 Samples per decade

Table 4.8 shows the impact of the *Samples per decade* parameter. The number of *Samples per decade* is shown in the first column, the hazard rate of the reduced model and the time of the reduction in the second and the third column respectively. The *minStep* is 0.1% during this measurement.

The plot in Fig. 4.16 shows the difference between the samples rate of failure distribution

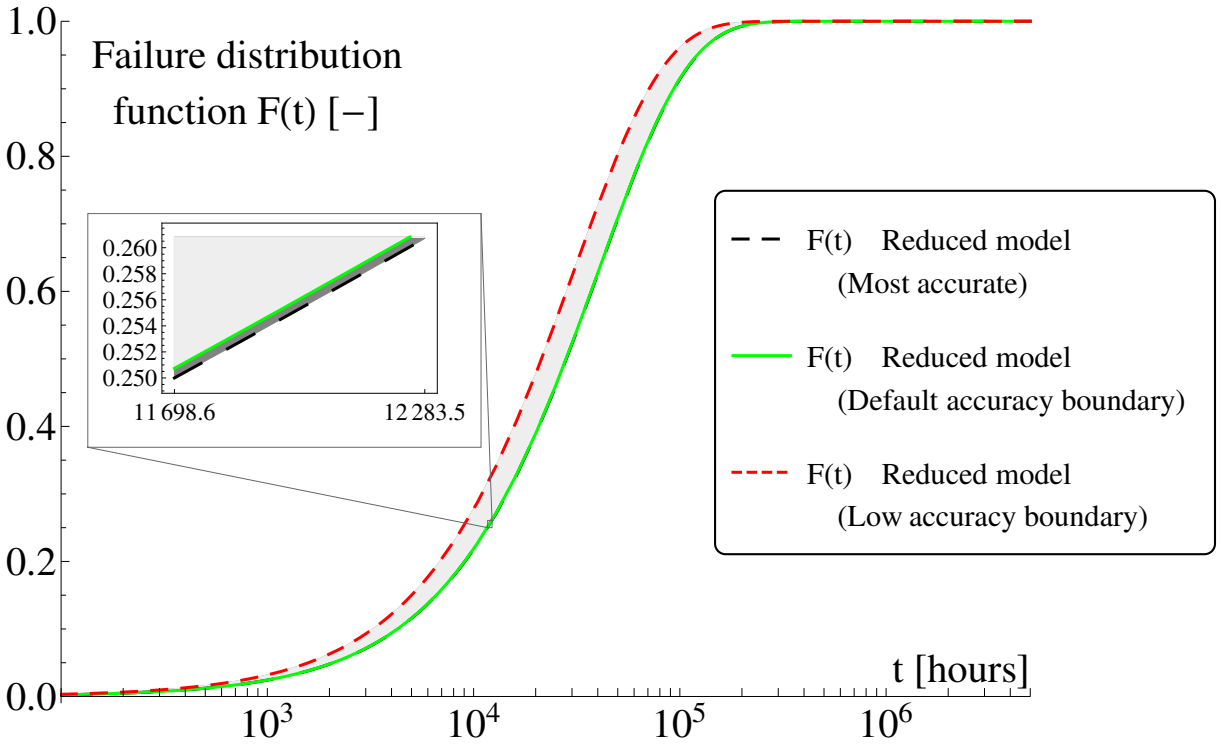

Figure 4.15: Comparison of failure distribution functions using the low and the default accuracy.

Table 4.8: Comparison of hazard rates and reduction times with respect to number of samples per each decade.

Samples per decade	$\lambda_{\mathbf{Hazard}}$ $[\times \mathbf{10^{-6}\,h^{-1}}]$	**Reduction time [s]**
10	22.852	0.08314
30	24.585	0.1061
100[1)]	**24.650**	**0.1887**
300	24.661	0.4157
1,000	24.664	1.3099
3,000	24.672	4.3333
10,000	24.674	17.537

[1)] Default value used in this thesis.

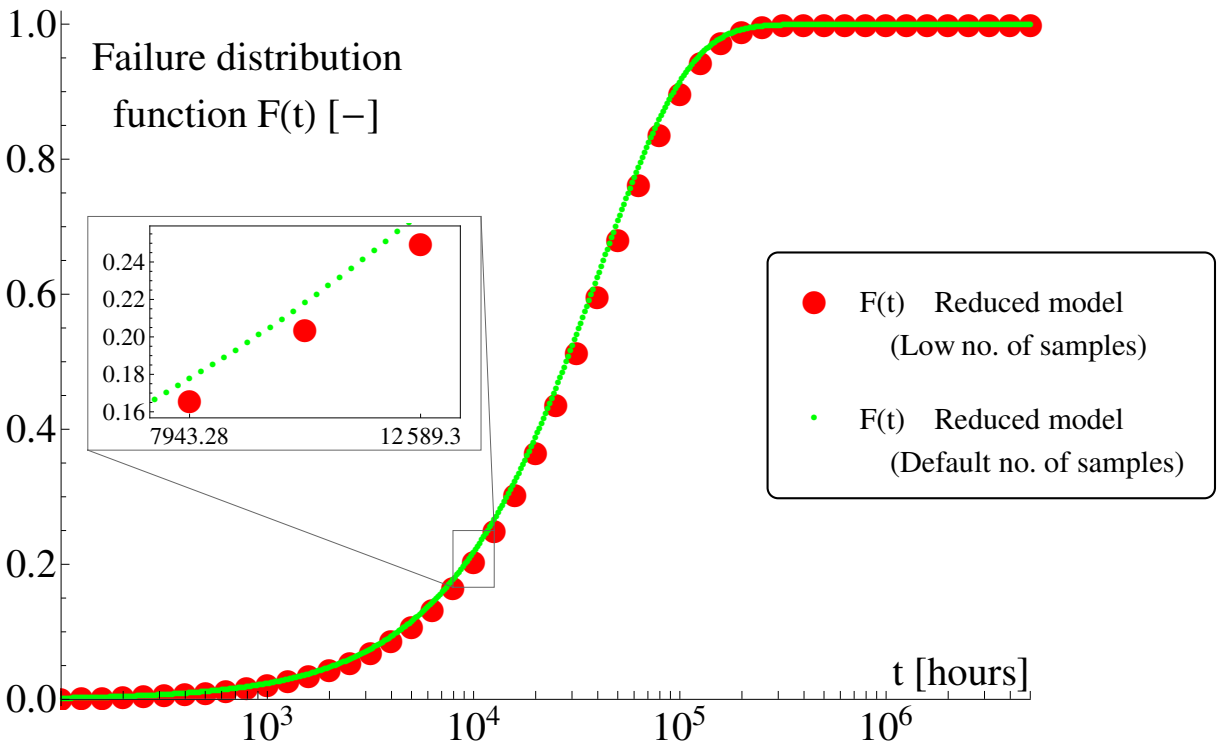

Figure 4.16: Comparison of failure distribution functions using the low and the default number of *Samples per decade*.

functions using the worst (10) *Samples per decade* value shown in Table 4.8 and the default value. The horizontal axis of the plot represents the time of operation measured in hours, the vertical axis represents the failure distribution function values. The small green dots represent the samples of the failure distribution function using the default *Samples per decade* value, the larger red dots represent the samples using the worst value.

The more *Samples per decade* are used, the higher (more precise) is the calculated hazard rate.

The reduction time is increasing linearly with respect to the number of *Samples per decade*, but the calculated hazard rate seems to be constant above 100 *Samples per decade* in this case.

The calculated hazard rate value is less pessimistic than the optimal value, but the difference between the hazard rates calculated using 30 and 10,000 *Samples per decade* is ca. 0.4%.

4.5 Application to Track Circuit System

The reduction has been used to calculate MTTF of a safety-critical Track Circuit System (TCS) developed by AZD Praha [33]. The aim of the TCS is to detect the presence of the

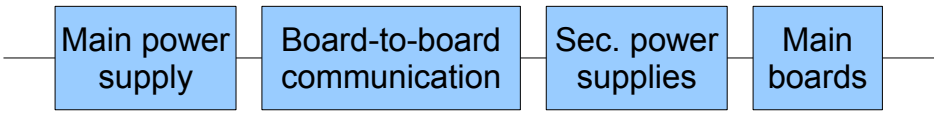

Figure 4.17: Top-level reliability block diagram of the Track Circuit System.

train and eliminate the possibility of a train collision. The TCS is a system consisting of three independent hot-swap computation modules. The implementation details of these modules are the intellectual property of AZD Praha, but they are not necessary for the purpose of this thesis. The calculation of MTTF of this system has been a project performed in our department. The creation and reduction of dependability models has been the author's participation in this project.

The model of the TCS uses three-level full reduction. The partial reduction cannot be used, because the system's developer is unable to guarantee the preventive maintenance of the system necessary to partial reduction. The model is heterogeneous: the first (lowest) and the third (top) level of the model are based on reliability block diagrams, the second level is based on Markov chains.

The main blocks of the TCS are: Main power supply, secondary power supplies (SPSs), main processing boards, and an internal checking unit (ICU). These main parts form the top level RBD model shown in Fig. 4.17. The failure of the ICU can hide the failure of the other parts of the system, but it cannot cause a critical failure of the system, thus it is not included in the RBD. The analysis of the system has shown a critical part of the board-to-board communication seriously affecting the total failure rate of the system, thus this part has been added to the top level RBD. The failure of the system means that the system is unable to provide its functionality and the traffic is stopped safely, thus the result of the reduction will be called λ_{Fail} is this section.

The SPSs are two independent hot-swap modules. Each SPS provides power to all processing boards and it is checked by the ICU. The Markov chain of the SPSs is shown in Fig. 4.18. The model assumes that all faults are detected instantly.

$Fault\_Free$ is the functional/fault-free state of the block. The failure rate of the first SPS is $2\lambda_{SPS}$, because the first fault can affect any of the two SPSs.

The $Fail_1$ state is active when one of the SPSs fails. This situation is detected by the ICU unit and reported. The SPSs remain in this state until the repair is finished (repair rate μ). The SPSs are fully functional in this state. The arc leading to $Fail$ state (rate λ_{SPS}) expresses the probability that a second SPS fails before the repair is finished.

If the ICU fails, the SPSs are locked in $Fail_{ICU}$ state. The SPSs are fully functional in this state, but a failure of an SPS (rate $2\lambda_{SPS}$) cannot be detected.

The $Fail_{ICU\_1}$ state is active when the ICU and one of the SPSs fails. This situation is not reported, thus the repair is not started and the SPSs remain in this state until a second SPS fails (rate λ_{SPS}).

The main processing boards are three independent boards. Each board performs the

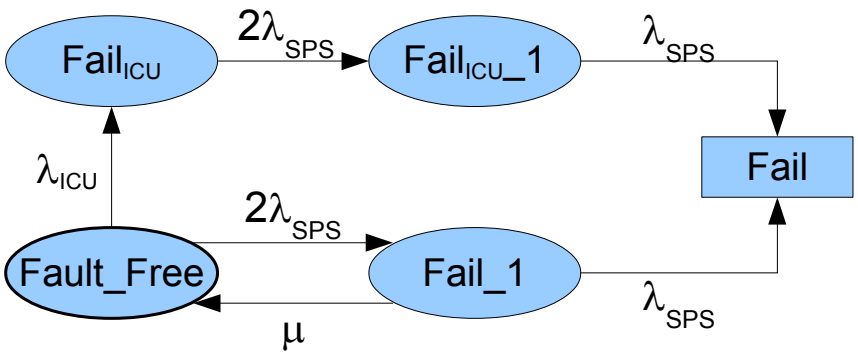

Figure 4.18: Markov chain of secondary power supplies of the Track Circuit System.

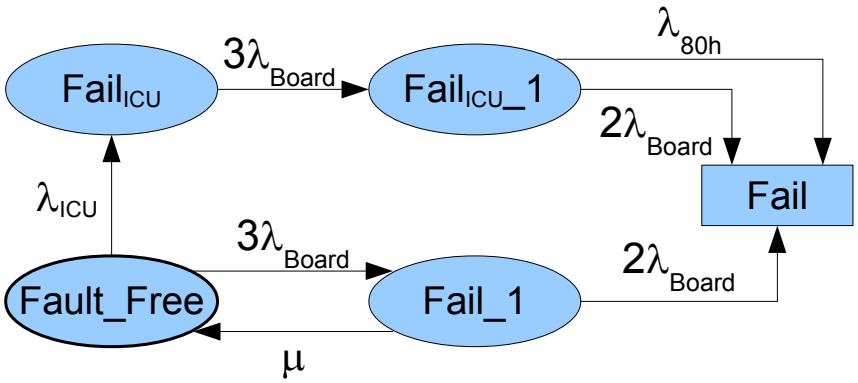

Figure 4.19: Markov chain of processing boards of the Track Circuit System.

same calculations independently of the other boards, but the results are compared to each other. Each board is checked by the ICU, too. Two fully functional boards are required to keep the system fully operational. The Markov chain of the boards is shown in Fig. 4.19. The model assumes that all faults are detected instantly.

$Fault\_Free$ is the functional/fault-free state of the block. The failure rate of the first board is $3\lambda_{Board}$, because the first fault can affect any of the three boards.

The $Fail_1$ state is active when one of the boards fails. This situation is detected by the ICU unit and reported. The boards remain in this state until the repair is finished (repair rate μ). The boards are fully functional in this state. The arc leading to $Fail$ state (rate $2\lambda_{Board}$) expresses the probability that a second board fails before the repair is finished.

If the ICU fails, the boards are locked in $Fail_{ICU}$ state. The boards are fully functional in this state, but a failure of a board (rate $3\lambda_{Board}$) cannot be detected.

The $Fail_{ICU\_1}$ state is active when the ICU and one of the boards fail. This situation is not reported, but the other boards starts an 80-hour countdown. A failure of the second board (rate λ_{Board}) or the finish of the countdown (rate λ_{80h}) causes the system to fail.

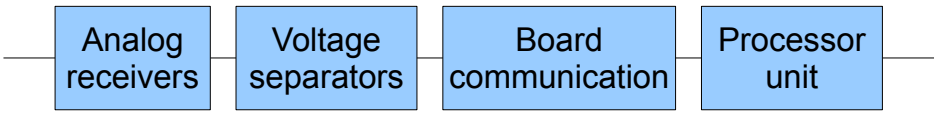

Figure 4.20: Low-level reliability block diagram of the Track Circuit System.

The countdown is also started in $Fail_1$ state, but the developer assumes that the repair of the system will be performed before the countdown is finished.

Each board is composed of the following blocks: Analog receivers collecting the informations from the rails, voltage separators not allowing one voltage failure to disable all boards simultaneously, the board communication (including a non-critical part of the board-to-board communication), and the processor unit. These parts form the low level RBD model shown in Fig. 4.20.

The failure rate of the atomic blocks of the model (all blocks of the low-level model, main power supply, and board-to-board communication blocks of the top-level model) are calculated using Parts Stress method according to standard MIL-HDBK-217F Notice2 [3]. The parameters of the environment follow:

Temperature: 40 °C

Environment type: Ground fixed – G_F

The failure rates of the low-level blocks:

Analog receivers: $29.99 \times 10^{-6}\,h^{-1}$

Voltage separator: $1.272 \times 10^{-6}\,h^{-1}$

Board communication: $10.31 \times 10^{-6}\,h^{-1}$

Processor unit: $10.14 \times 10^{-6}\,h^{-1}$

$\lambda_{Board} = 51.72 \times 10^{-6}\,[h^{-1}]$

The parameters of the Markov chain of the processing boards:

$\mu = 1/40\,[h^{-1}]$ – the repair rate

$\lambda_{ICU} = 1.810 \times 10^{-6}\,[h^{-1}]$ – the failure rate of the ICU

$\lambda_{80h} = 1/80\,[h^{-1}]$ – the inverted countdown delay

$\lambda_{Main\_boards} = 2.435 \times 10^{-6}\,[h^{-1}]$

The parameters of the Markov chain of the secondary power supplies:

$\mu = 1/40\,[h^{-1}]$ – the repair rate

$\lambda_{SPS} = 19.01 \times 10^{-6}\,[h^{-1}]$ – the failure rate of one secondary power supply

$\lambda_{ICU} = 1.810 \times 10^{-6}\,[h^{-1}]$ – the failure rate of the ICU

$\lambda_{Sec.\_Power\_Supp.} = 1.826 \times 10^{-6}\,[h^{-1}]$

The failure rates of the top-level blocks:

Main power supply: $0.287 \times 10^{-6}\,h^{-1}$

Board-to-board communication: $12.41 \times 10^{-6}\,h^{-1}$

$\lambda_{Main\_boards} = 2.435 \times 10^{-6}\,[h^{-1}]$

$\lambda_{Sec.\_Power\_Supp.} = 1.826 \times 10^{-6}\,[h^{-1}]$

$\lambda_{Fail} = 16.96 \times 10^{-6}\,[h^{-1}]$

$$MTTF = \frac{1}{\lambda_{Fail}} = 58{,}962\,[h]\,(= 6.7\,years)$$

The plot in Fig. 4.21 shows the comparison of the failure distribution functions of the TCS. The horizontal axis of the plot represents the time of operation measured in hours, the vertical axis represents the failure distribution function values. The black long-dashed line represents the exact model failure distribution function and the green line represents the reduced model failure distribution function calculated using the hierarchy model. The red short-dashed line shows the relative error between the exact and the reduced failure distribution functions.

The relative error between the exact and the reduced failure distribution functions is low – it does not exceed 35%, even though the full reduction is used.

The board-to-board communication contains a critical part. The failure of this part leads to the failure of the whole system (the processor units are unable to compare their data, thus they will stop the traffic immediately). If this part is altered to be more safe (to contain several independent communication lines), the processor units will be able to communicate using the unaffected lines.

This alternation will allow *Board-to-board communication* block to be moved from the top-level model to the low-level one. The failure rate is such altered system would be $\lambda_{Fail\_Altered} = 4.885 \times 10^{-6}\,h^{-1}$, thus $MTTF_{Altered} = 204{,}709\,[h]$ (ca. 23 years). This alternation would be a significant improvement to the MTTF of the system.

4.6 Summary

The reduction of hierarchical dependability models based on Markov chains has been presented. It has been used to efficiently calculate the hazard rates of safety-critical systems. Two presented case studies (2oo2-NMR and MDS-NMR) use two-level dependability

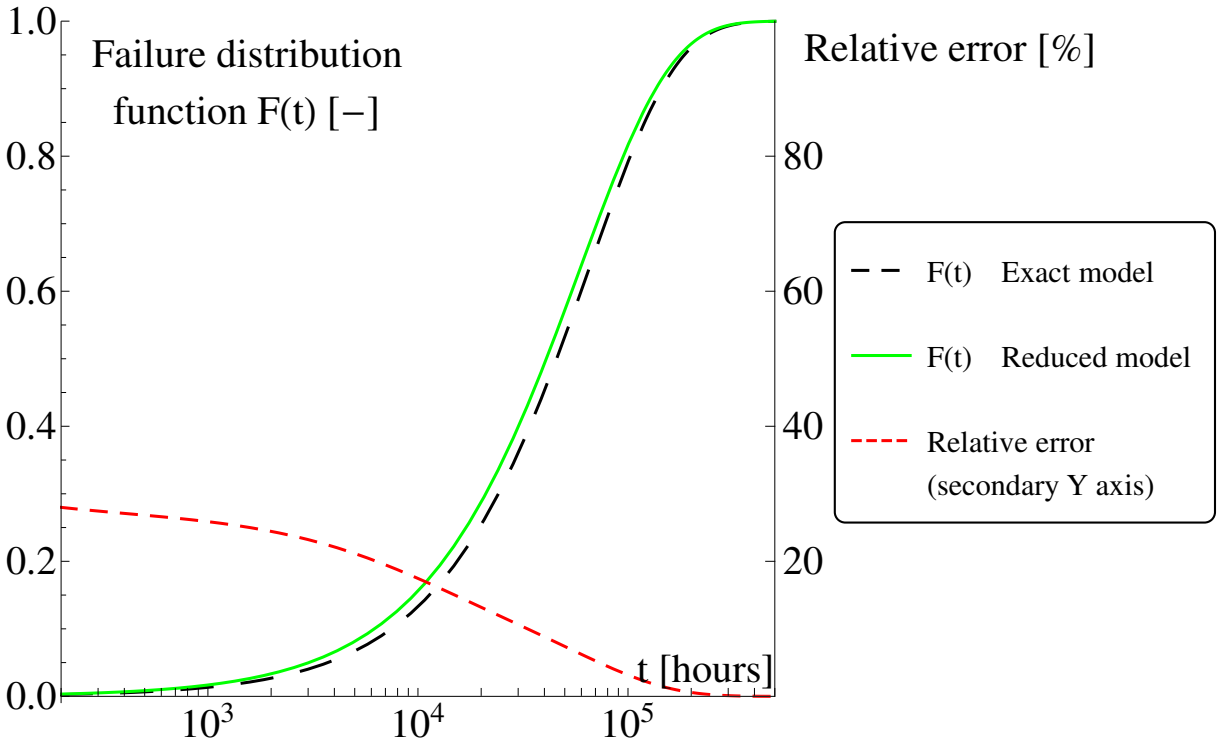

Figure 4.21: Comparison of failure distribution functions of actual dependable system.

models based on Markov chains, the third (TCS) uses three-level dependability models based on Markov chains and reliability block diagrams.

X-NMR case studies are based on the Two-out-of-two model or the Modified duplex system model as the low-level model and the N-modular redundancy as the high-level model. Both low-level models are related to the railway interlocking equipment. The current relay-based ones are modeled using the first one, the second one is about to be used in an FPGA-based replacements of these equipments.

The results indicate that the hazard rates of the hierarchical models can be calculated significantly faster using the presented reduction, compared to the calculation using the exact model. This speedup is the main contribution of the reduction. The failure distribution functions of the low-level and the high-level model are solved separately, thus the reduction time does not depend on the the number of the low-level blocks. On the other hand, the calculation time of the exact model grows exponentially with the number of low-level blocks.

The main disadvantage of the presented reduction is its inaccuracy. The failure distribution functions of the exact model and the reduced models extremely differ when the probability of the failure of the modeled system is low. This problem can be fixed using the partial reduction method. This method offers a significantly more accurate solution, but it is not pessimistic all the time, thus it can be used only if the preventive maintenance of the system is performed before the non-pessimistic time is reached.

The partial reduction allows a trade-off between the hazard rate (SIL) and the preventive maintenance period of the system to be found. Three types of the partial reduction are proposed:

- *Time-limited reduction* – uses t_{limit} itself

  The most direct approach. The hazard rate of the system calculated using this method is lower than the hazard rate calculated using the full reduction. This value is valid only until the end of the preventive maintenance period given by t_{limit} is met.

  If the maintenance period is 200,000 hours of operation (ca. 22 years), the hazard rate of the presented case study system is ca. $5 \times 10^{-7}\,h^{-1}$ and it will be classified as SIL2 (the hazard rate of the same system without maintenance period specified would be ca. $25 \times 10^{-6}\,h^{-1}$ and the system would not meet the requirements to be classified as SIL1). If the maintenance period is 150,000 hours of operation (ca. 17 years), the hazard rate can be decreased to ca. $6 \times 10^{-8}\,h^{-1}$ that is close to meet the SIL4 requirement suitable for the railway station signaling and interlocking equipment and other safety-critical systems.

- *Probability-limited reduction* – uses a p_{limit} probability

  This approach is similar to the first one. It also leads to lower hazard rate of the system valid only until the end of the maintenance period. This end is given by p_{limit}. If the total probability of failure of the system reaches p_{limit}, the maintenance period ends.

- *Hazard-rate-limited reduction* – uses a λ_{limit} hazard rate

  This approach slightly differs form the previous ones. The target hazard rate is given by λ_{limit} or by SIL and the end of the maintenance period is calculated as the point, where the reduced failure distribution function stops being pessimistic.

  The presented case study can meet SIL2 requirements, when the maintenance period is ca. 234,000 hours (ca. 26 years), SIL3 can be met with period ca. 162,000 hours (ca. 18 years), and SIL4 can be met with period ca. 126,000 hours (ca. 14 years). Note, that neither the system nor the fault rate has been changed to meet significantly better SIL requirements. If the system meets SIL4 requirements, it can be used as a safety-critical system.

The reduction is not limited to hazard rate or SIL calculations. It can be used to calculate mean time to failure of the system, too. MTTF of the TCS case study has been calculated using three-level dependability models based on Markov chains (the second level) and reliability block diagrams (the first and the top levels). The reduction allows a simple cooperation of both types of models (the hazard rates calculated using one model can be easily used as the parameter of the other model and vice versa). MTTF of the TCS is ca. 60,000 hours (ca. 6.7 years), but the model has been used also to predict MTTF of the altered system using more dependable board-to-board communication. MTTF of the altered TCS will be ca. 200,000 hours (ca. 23 years). TCS case study also show, that the

inaccuracy of the full reduction does not need to be unsuitable for actual systems. The relative error between the exact and the reduced failure distribution functions is low – it does not exceed 35%, even though the full reduction is used.

Chapter 5

Conclusions

5.1 Summary

This dissertation thesis presents simplified dependability models and methods for easier dependability parameters computations. These models based on absorbing Markov chains are able to model (self-)repairing capabilities and they can be used to create a hierarchical dependability model of a system, thus they allow dependability parameters of large and complex systems to be calculated without the state-explosion issues.

The proposed method also allows an absorbing Markov chain to be used as a block/event of any type of dependability model, that can be used to calculate a hazard rate (reliability block diagrams, stochastic Petri nets, dynamic fault trees, etc.).

The key step to create hierarchical dependability models based on Markov chains is the proposed reduction method. The reduction is inexact, but pessimistic method allowing the hazard rate of the block/system to be calculated. The partial reduction presented in Section 3.2 is able to provide significantly more accurate results, but it can be used only when the preventive maintenance is guaranteed, i.e. the modeled system will be replaced/repaired before the end of the maintenance period of the system is reached. The required period of the preventive maintenance is the result of the partial reduction, too.

The reduction and the principle of the hierarchical dependability models are applied on the case study systems in Chapter 4. The dependable blocks used in the first system use Two-out-of-two redundancy. The second system is based on blocks using Modified duplex system redundancy. Both types of redundancies are related to dependability models of the railway station signaling and interlocking equipment. The results show that the hazard rate of the system can be calculated significantly faster using the presented reduction. The speedup is important when the system contains many dependable blocks, because the calculation of hazard rate using the non-hierarchical model of such system can become computationally impossible in practice. The results also show the main disadvantage of the reduction – the inaccuracy of the solution may not allow the results to be used in practical applications.

The partial reduction shown in Section 4.3 is used to resolve the inaccuracy of the

results. The same system with the same fault rate can achieve significantly better hazard rates when the partial reduction is used. The partial reduction also allows a calculation leading to guaranteed levels of dependability parameters at the cost of the reduced maximal allowed operational time of the system.

The reduction has been also applied on the actual dependable track circuit system in Section 4.5 to calculate the mean time to failure of the system. A simple cooperation of several types of models (the hazard rates calculated using one model can be easily used as the parameter of the other model and vice versa) has been presented. The model has been used also to predict MTTF of the altered system using more dependable board-to-board communication. The track circuit system case study also shows, that the inaccuracy of the full reduction does not need to be unsuitable for actual systems, because the relative difference is low, even though the full reduction is used.

5.2 Contributions of the Thesis

- The heterogeneous hierarchical dependability models allowing multiple types of dependability models to be used in a model of a complex system.
- The dependability models reduction allowing inexact, pessimistic dependability parameters calculations to be performed in a few seconds – even in the case of large complex systems, where the results of classical detailed models are practically unreachable due to state explosion.
- The partial reduction allowing calculation leading to guaranteed levels of dependability parameters at the cost of the reduced maximal allowed operational time of the system (mandatory preventive maintenance with the period calculated during the partial reduction).
- Experimental verification of the presented methods on the model of the complex system based on the real models of the railway interlocking equipment. The experiments have shown, that the calculation time of the hierarchical model does not depend exponentially on the number of its blocks. They also shown, that the partial reduction can be used to calculate the preventive maintenance period of the system, when the requirement of its hazard rate is strictly defined (e.g. by international standards in the case of the railway equipment safety-critical systems).

5.3 Future Work

The author of the dissertation thesis suggests to explore the following:

- The analysis of conditions leading to the nearly-exponential shape of the exact failure distribution function of the system should be performed. This analysis could provide us information, what kind of systems can be reduced using the full method without

inaccuracy issues, and what kind of systems should be reduced using the partial method.

- The accuracy of the partial reduction could be improved by multiple limit values. The improved method could provide us information, how the hazard rate depends on the preventive maintenance period of the system.
- The further increase of the number of the limit values (a limit value for each failure distribution function sample) could allow us to calculate with variable hazard rates of the blocks. This improvement could help us to overcome one of the disadvantages of Markov chains – the requirement of constant intensity rates of all events modeled by a Markov chain.
- The hierarchical dependability models will be applied to the FPGA-based safety-critical systems with realistic fault models based on realistic radiation exposure experiments performed on real FPGA chips currently performed by colleagues from our department.

Bibliography

[1] *Electronic Reliability Design Handbook – MIL-HDBK-338B*. US Department of Defense, 1998.
Available from: `http://www.weibull.com/mil_std/mil_hdbk_338b.pdf`

[2] Vesely, W. *Fault Tree Handbook with Aerospace Applications*. National Aeronautics and Space Administration (NASA), 2002.
Available from: `http://www.hq.nasa.gov/office/codeq/doctree/fault_tree.htm`

[3] *Reliability Prediction of Electronic Equipment – MIL-HDBK-217F Notice 2*. US Department of Defense, 1995.
Available from: `http://www.weibull.com/mil_std/mil_hdbk_217f_2.pdf`

[4] System Reliability Center – PRISM.
Available from: `https://src.alionscience.com/prism/`

[5] Denson, W. *Handbook of 217Plus Reliability Prediction Models*. Reliability Information Analysis Center (RIAC), 2006, ISBN 9781933904023.

[6] European Standards EN 50129:2003 – Railway applications: Communication, signalling and processing systems: Safety-related electronic systems for signalling.

[7] Avizienis, A.; Laprie, J.-C.; Randell, B.; et al. Basic concepts and taxonomy of dependable and secure computing. *Dependable and Secure Computing, IEEE Transactions on*, volume 1, no. 1, Jan 2004: pp. 11–33, ISSN 1545-5971, doi:10.1109/TDSC.2004.2.

[8] Kirrmann, H. *Fault Tolerant Computing in Industrial Automation*. ABB Research Center, second edition, 2005.

[9] Shooman, M. L. *Reliability of Computer Systems and Networks: Fault Tolerance,Analysis,and Design (Appendix B: Summary of Reliability Theory)*. New York, NY, USA: John Wiley & Sons, Inc., 2002, ISBN 0471293423.

[10] O'Connor, P.; Kleyner, A. *Practical Reliability Engineering*. Wiley Publishing, fifth edition, 2012, ISBN 047097981X, 9780470979815.

[11] Wikipedia – Safety. modified on 14 August 2015.
Available from: `http://en.wikipedia.org/wiki/Safety`

[12] Shooman, M. L. *Reliability of Computer Systems and Networks: Fault Tolerance,Analysis,and Design (Appendix A: Summary of Probability Theory)*. New York, NY, USA: John Wiley & Sons, Inc., 2002, ISBN 0471293423.

[13] Lanus, M.; Yin, L.; Trivedi, K. Hierarchical composition and aggregation of state-based availability and performability models. *Reliability, IEEE Transactions on*, volume 52, no. 1, March 2003: pp. 44–52, ISSN 0018-9529, doi:10.1109/TR.2002.805781.

[14] Petri, C. A. *Communication with automata*. Dissertation thesis, Universität Hamburg, 1966.

[15] Marsan, M. A. Stochastic petri nets: An elementary introduction. In *In Advances in Petri Nets*, Springer, 1989, pp. 1–29.

[16] Murata, T. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, volume 77, no. 4, Apr 1989: pp. 541–580, ISSN 0018-9219, doi:10.1109/5.24143.

[17] Bause, F.; Kritzinger, P. S. *Stochastic Petri Nets: An Introduction to the Theory*. Vieweg Verlag, second edition, 2002, ISBN 3-528-15535-3.

[18] Esparza, J. Decidability and complexity of Petri net problems An introduction. In *Lectures on Petri Nets I: Basic Models*, *Lecture Notes in Computer Science*, volume 1491, edited by W. Reisig; G. Rozenberg, Springer Berlin Heidelberg, 1998, ISBN 978-3-540-65306-6, pp. 374–428, doi:10.1007/3-540-65306-6_20.

[19] Esparza, J. Reachability in live and safe free-choice Petri nets is NP-complete. *Theoretical Computer Science*, volume 198, no. 12, 1998: pp. 211–224, ISSN 0304-3975, doi:http://dx.doi.org/10.1016/S0304-3975(97)00235-1.

[20] Praveen, M. *Complexity of the Reachability Problem in Subclasses of Petri Nets*. Master's thesis, The Institute of Mathematical Sciences, 2008.
Available from: `http://www.imsc.res.in/~praveen/theses/MSCThesis.pdf`

[21] Malhotra, M.; Trivedi, K. Dependability modeling using Petri-nets. *Reliability, IEEE Transactions on*, volume 44, no. 3, Sep 1995: pp. 428–440, ISSN 0018-9529, doi:10.1109/24.406578.

[22] Marsan, M. A.; Balbo, G.; Conte, G.; et al. *Modelling with Generalized Stochastic Petri Nets*. New York, NY, USA: John Wiley & Sons, Inc., first edition, 1994, ISBN 0471930598.

[23] Guo, H.; Yang, X. A simple reliability block diagram method for safety integrity verification. *Reliability Engineering & System Safety*, volume 92, no. 9, 2007: pp. 1267–1273, ISSN 0951-8320, doi:http://dx.doi.org/10.1016/j.ress.2006.08.002.

[24] Haasl, D. F. Advanced Concepts in Fault Tree Analysis. In *Proceedings of the System Safety Symposium*, 1965.

[25] Bechta Dugan, J.; Bavuso, S. J.; Boyd, M. Dynamic fault-tree models for fault-tolerant computer systems. *Reliability, IEEE Transactions on*, volume 41, no. 3, Sep 1992: pp. 363–377, ISSN 0018-9529, doi:10.1109/24.159800.

[26] Chiacchio, F.; Cacioppo, M.; D'Urso, D.; et al. A Weibull-based compositional approach for hierarchical dynamic fault trees. *Reliability Engineering & System Safety*, volume 109, 2013: pp. 45–52, ISSN 0951-8320, doi:http://dx.doi.org/10.1016/j.ress.2012.07.005.

[27] Bechta Dugan, J. Fault trees and imperfect coverage. *Reliability, IEEE Transactions on*, volume 38, no. 2, Jun 1989: pp. 177–185, ISSN 0018-9529, doi:10.1109/24.31102.

[28] Anand, A.; Somani, A. Hierarchical analysis of fault trees with dependencies, using decomposition. In *Reliability and Maintainability Symposium, 1998. Proceedings., Annual*, Jan 1998, ISSN 0149-144X, pp. 69–75, doi:10.1109/RAMS.1998.653591.

[29] Dobiáš, R.; Kubátová, H. The Common 2oo2 Safety Model for Signalling and Interlocking Equipments. In *Electronic Circuits and Systems Conference*, Slovak University of Technology, 2005, pp. 81–84.

[30] Dobiáš, R.; Kubátová, H. FPGA based design of the railway's interlocking equipments. In *Digital System Design, 2004. DSD 2004. Euromicro Symposium on*, Aug 2004, pp. 467–473, doi:10.1109/DSD.2004.1333312.

[31] Kubalík, P.; Kubátová, H. Dependable design technique for system-on-chip. *Journal of Systems Architecture*, volume 54, no. 34, 2008: pp. 452–464, ISSN 1383-7621, doi:http://dx.doi.org/10.1016/j.sysarc.2007.09.003.

[32] Wolfram Mathematica web page.
Available from: `http://www.wolfram.com/mathematica/`

[33] AZD Praha s.r.o. web page.
Available from: `http://www.azd.cz/`

Reviewed Publications of the Author Relevant to the Thesis

[A.1] Kohlík, M.; Kubátová, H. Reduction of Complex Safety Models based on Markov Chains. In *Proceedings of the 2012 IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits and Systems (DDECS)*, pp. 183–186, Tallinn, Estonia, 2012.

[A.2] Kohlík, M.; Kubátová, H. Markov chains hierarchical dependability models: Worst-case computations. In *14th Latin American Test Workshop*, Cordoba, Argentina, 2013.

[A.3] Kohlík, M.; Kubátová, H. Hierarchical Models of Markov Chains: Optimizations with Limited Pessimism. In *Proceedings of the 18th International Conference Electronics 2014*, pp. 59–62, Palanga, Lithuania, 2014.

[A.4] Kohlík, M.; Kubátová, H. Hierarchical Dependability Models Based on Pessimistic Reduction of Markov Chains. To be published in *Reliability Engineering & System Safety* (draft accepted, major revision required).

Remaining Publications of the Author Relevant to the Thesis

[A.5] Kohlík, M. Dependability models based on Petri nets and Markov chains. In *Počítačové architektury & diagnostika*, pp. 95–103, Soláň, Czech Republic, 2009.

[A.6] Kohlík, M. Dependability models for reconfigurable FPGA modular design. *A Doctoral Study Report*, Faculty of Information Technology, Prague, Czech Republic, 2010.

[A.7] Kohlík, M.; Kubátová, H. Hierarchical Dependability Models Based on Markov Chains. In *Počítačové architektury & diagnostika*, pp. 145–150, Milovy, Czech Republic, 2012.

[A.8] Kohlík, M.; Kubátová, H. Hierarchical Dependability Models Based on Markov Chains. In *Proceedings of 2013 26th International Conference on Architecture of Computing Systems (ARCS)*, Prague, Czech Republic, 2013.

Remaining Publications of the Author

[A.9] Kohlík, M.; Kubátová, H. Reconfiguration Strategy for FPGA Dependability Characteristics Improvement based on Stochastic Petri Net. In *Proceedings of 4th Descrete-Event System Design*, pp. 253–257, Valencia, Spain, 2009.

[A.10] Borecký, J.; Kohlík, M.; Kubátová, H.; Kubalík, P. Faults Coverage Improvement based on Fault Simulation and Partial Duplication. In *Proceedings of the 13th Euromicro Conference on Digital System Design*, pp. 380–386, Lille, France, 2010.

[A.11] Kohlík, M.; Kubátová, H. Model of Modular Secured Designs for Calculations of Availability. In *Proceedings of the Work in Progress Session SEAA 2010 and DSD 2010*, pp. 15–16, Lille, France, 2010.

[A.12] Borecký, J.; Kohlík, M.; Kubátová, H. How to Measure Dependability Parameters of Programmable Digital Circuits – A Survey. In *6th Doctoral Workshop on Mathematical and Engineering Methods in Computer Science*, pp. 28–35, Mikulov, Czech Republic, 2010.

[A.13] Borecký, J.; Kohlík, M.; Kubalík, P.; Kubátová, H. Fault Models Usability Study for On-line Tested FPGA. In *Proceedings of the 14th Euromicro Conference on Digital System Design*, pp. 287–290, Oulu, Finland, 2011.

[A.14] Borecký, J.; Kohlík, M.; Kubátová, H. Miscellaneous Types of Partial Duplication Modifications for Availability Improvements. In *Proceedings of the 15th Euromicro Conference on Digital System Design*, pp. 79–83, Izmir, Turkey, 2012.

[A.15] Vít, P.; Borecký, J.; Kohlík, M.; Kubátová, H. Fault Tolerant Duplex System with High Availability for Practical Applications. In *Proceedings of the 17th Euromicro Conference on Digital System Design*, pp. 320–325, Verona, Italy, 2014.