Markov Chains Hierarchical Dependability Models: Worst-case Computations

Martin Kohlík and Hana Kubátová
Department of Digital Design
Faculty of Information Technology
Czech Technical University in Prague
Thákurova 9, 160 00 Prague 6, Czech Republic
{martin.kohlik, hana.kubatova}@fit.cvut.cz

Abstract—**Dependability models allow calculating the rate of an event 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. A hierarchical dependability model allows expressing multiple redundancies made at multiple levels of a system decomposed to multiple cooperating blocks. A hierarchical dependability model based on Markov chains allows each block and relations between these blocks to be expressed independently by Markov chains. This allows a decomposition of a complex dependability model into multiple small models to be made. The decomposed model is easier to read, understand and modify. A hazard rate is calculated significantly faster using hierarchical model, because the decomposition allows exponential calculation-time explosion to be avoided. The paper shows a method how to reduce Markov chains and use them to create hierarchical dependability models. An example study is used to demonstrate the advantages of the hierarchical dependability models (the decomposition of the complex model into multiple simple models and the speedup of the hazard rate calculation).**

I. Introduction

The design methods of dependable systems that can be used in mission-critical applications is the area of interest of our research group in Faculty of Information Technology (FIT). We have been using field-programmable gate arrays (FPGAs) in such practical applications as railway station signaling and interlocking equipment, automotive systems, etc., due to their flexibility and ”time to market”. But FPGA-based systems are sensitive to many effects that can change their programmed function [1]. These changes are most unwelcome in dependable systems, where financial losses, serious injuries or casualties can be caused because of a failure. Therefore the appropriate methods of dependability modeling, which will guarantee the worst-case dependability parameters and which will be able to perform these computations as simple and as quick as possible is needed.

Dependability is an integrating concept including Availability, Safety, Reliability, Integrity and Maintainability. We focus on the Safety parameter defined as absence of catastrophic consequences on the user(s) and the environment [2].

One of the most important design techniques allowing improvement of dependability is redundancy. This means that if one part of the system fails, there is an alternate functional part. However, redundancy can have a negative impact on a system performance, size, weight, power consumption, and others [3].

There are many redundancy techniques including hardware, information, time, software redundancy etc. [3]. We use hardware redundancy made by replication in this paper.

Hierarchical dependability models are especially suitable, when the system is built from multiple dependable blocks. Each dependable block can use internal redundancy and another level of redundancy can be used outside the blocks to greatly reduce the probability of safety violation.

An event causing violation of safety of a system will be called *hazard event*. The rate of a hazard event is called *hazard rate*.

Dependability models – models designed to calculate a hazard rate of a system – may be created as exact or inexact models. Models of complex systems composed of cooperating blocks may be created as coarse-grained – small and simple models allowing exact calculations of hazard rate in a short time. On the other hand, coarse-grained models are inaccurate and do not reflect the internal structure of the system. Fine-grained models are accurate, but they can be too large, and thus the hazard rate calculation is time-consuming. They reflect the internal structure, but they grow rapidly when the complexity of a system – e.g. the number of the dependable blocks – is increased.

Inexact models may be used to speed up the calculations, because accuracy is not crucial, if we prove that the inexact result is pessimistic. In other words, we must prove that the real system will be safer than the system modeled by the inexact model(s).

This paper presents a method of reducing dependability models based on Markov chains that allows us to reduce low-level models, so they contain one transition with one hazard rate only. The transition corresponds to hazard event of modeled part of the system.

The reduction allows inexact hierarchical models to be built. They use multiple linked models to reflect the structure of a system. Multi-level hierarchy may be used to describe each level of redundancy independently. The hazard rates calculated from low-level models are used in higher-level models. Higher-level models are also reduced and their hazard rates are used in top-level models.

Hierarchical models are composed of multiple small models, so they are easier to read/understand, easier to modify/manipulate, and allow exponential calculation-time explosion to be avoided (the dependability parameters are calculated significantly faster).

The proposed reduction method is demonstrated on case study system containing multiple (up to 17) identical dependable blocks configured as an N-modular redundant system (NMR) in this paper. Hierarchical model is used to illustrate the reduction method. The hierarchical model uses 2 linked models (a top NMR model and a model of the block) containing up to 14 states in total, instead of up to tens of thousands states of the exact model that would result from the Cartesian product of all models.

The results indicate that the hazard rate calculated using the hierarchical model is higher than the hazard rate calculated without hierarchy, but the CPU-time spent on the reduction of the hierarchical model is greatly reduced (up to 80 times compared to the same system modeled by a standard non-hierarchical model).

The paper is organized as follows: Section II introduces basic reliability definitions. The reduction procedure is described in Section IV and applied on the case study system in Section V. Section VI concludes the paper.

II. Theoretical Background and Related Models

The failure distribution function $F(t)$ is a complementary function to reliability function. Reliability function $R(t)$ is a probability that the system will perform its intended function under specified design limits from time 0 until time t at least [2], [4].

Hazard rate (λ) is defined as a constant failure rate $f(t)$ that is a conditional probability of failure density function if the failure has not occurred until time t [4]. The hazard rate of the system is the key value to calculate the value of SIL.

The hierarchical models used to calculate the hazard rate of the system are based on non-renewable Markov chains ([5] Section 6.4). Markov chains (MCs) are able to model systems whose events are defined by discrete probability values and are also able to model systems whose events are defined by continuous intensity rates. The second variant is more suitable to determine SIL, because the value of SIL is based on hazard rate of the modeled system.

A non-renewable Markov chain contains hazard and non-hazard states. Hazard states represent situations where safety of the modeled system is violated. There are paths from each non-hazard state leading to a hazard state and there are no paths leading from a hazard state to a non-hazard state.

The main advantage of Markov chains is the simple calculation of reliability function of the modeled system using the system of differential equations when all events satisfy the Markovian property. The Markovian property is met, when the future of the modeled system is based on its present state only. Nowadays mathematical software is able to solve such system automatically using analytic or numeric methods. The main disadvantage of Markov chain is the state explosion – the number of the states of the model grows fast when modeled system complexity is increased.

The Fault Trees (FTs) and Reliability Block Diagrams (RBDs) are commonly used to build hierarchical dependability models [6], but there is one main disadvantage – elementary events/blocks of these models must be independent. Hierarchical dependability models based on MCs allow us to model any dependability among the events/blocks that can be modeled by an MC.

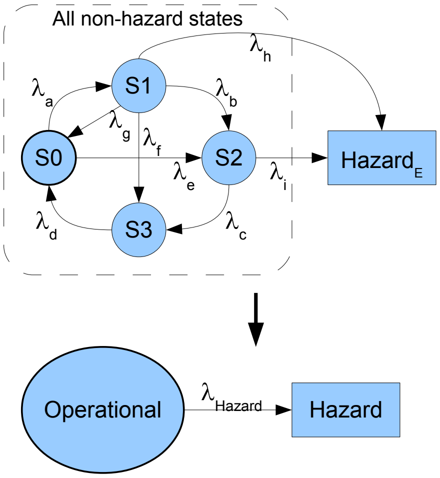

Fig. 1. Illustrative example of dependability model reduction.

Hierarchical dependability models based on FTs or RBDs can use MCs as the models of the events/blocks, when elementary events/blocks are independent. The well-known method shown in [6] uses the limit value of failure distribution function $\lim_{t\to\infty} F(t)$ (steady-state availability) of the MC in the upper level of hierarchy. The steady-state availability cannot be used to calculate hazard rate of the modeled system, because the hazard state of non-renewable MC will be reached sooner or later and there is no way back thus the steady-state availability of any non-renewable MC is always equal to 0.

III. Dependability Models Reduction

The reduction of a dependability model is made by joining all non-hazard states into a single state as shown in Fig. 1. The hazard state remains intact, the index ”E” (Exact) is used to mark the hazard state of the exact model. Any dependability model can be reduced using the reduction to the same model shown in the lower part of Fig. 1. The reduced model contains a new hazard rate λ_{Hazard} – the hazard rate substituting all hazard rates in the exact model.

The drawback of the reduction is the loss of accuracy, because the failure distribution functions of the exact and the reduced models are not equal. The constant hazard rate calculated from the reduced system leads to exponential failure distribution function, but the failure distribution function calculated from the exact system can have general shape.

Accuracy is not crucial in our case, but we must prove that the inaccurate hazard rate calculated from the reduced model is pessimistic. In other words, we must prove that the real system will be at least as safe as the system modeled by the reduced model(s). The proof is made by comparison of the reduced failure distribution function $F_R(t) = 1 - e^{(-\lambda_{Hazard} * t)}$ with the failure distribution function $F_E(t)$ of the exact model. The condition is called the **main requirement**. The main requirement is met when

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

In other words, the reduced failure distribution function must be greater than the failure distribution function of the exact model all the time.

The reduction is made as follows:

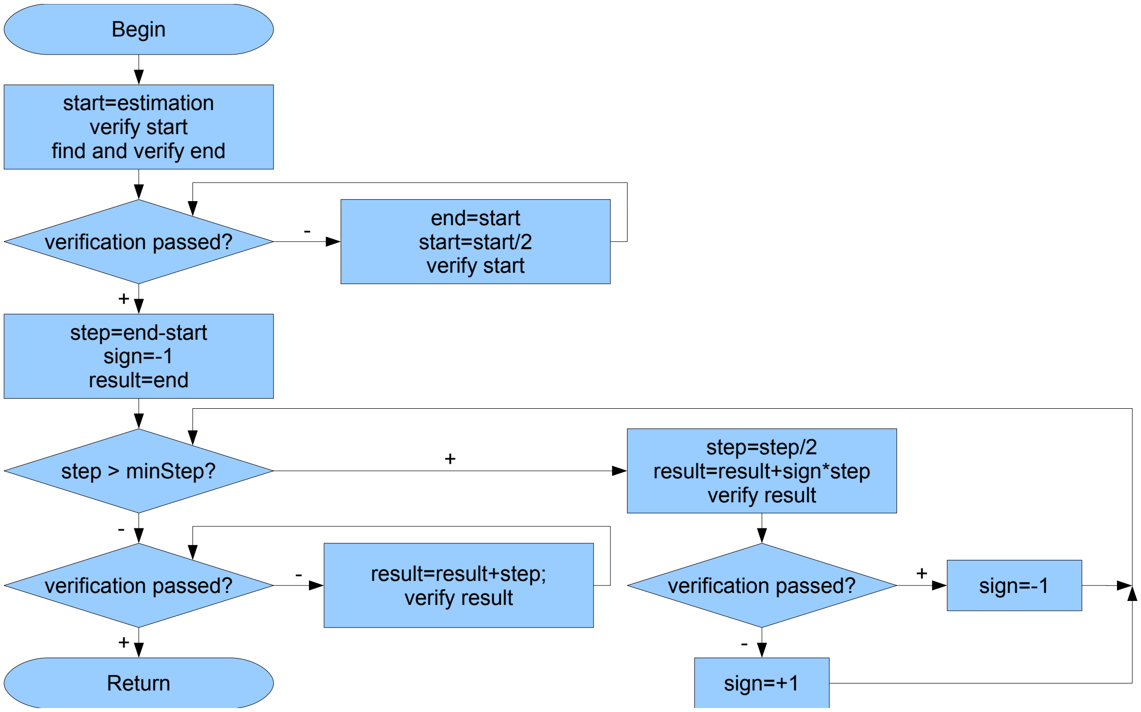

Fig. 2. Correction algorithm flowchart.

1) Calculate the exact failure distribution function $F_E(t)$[1].
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 of the next step. This value may not meet the main requirement, but the better is the estimation, the faster will be the next step.
   λ_{Hazard_Est} is taken from the estimated failure distribution function $F_R(t)$ intersecting the exact failure distribution function $F_E(t)$ from the previous step at the predefined probability *level*.

$$F_E(t_{level}) = F_R(t_{level}) = 1 - e^{(-\lambda_{Hazard\_Est} * t_{level})}$$

3) Make correction of λ_{Hazard_Est} to satisfy the main requirement.
   The goal of the correction is to find the lowest value of λ_{Hazard_Est} whose $F_R(t)$ meets the main requirement with the accuracy given by *minStep* parameter. 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 $F_E(t) \leq F_R(t)$ all the time and λ_{Hazard_Est} has the lowest possible value. There are three main loops (see the flowchart shown in Fig. 2):
   a) Find and verify the endpoints of the interval that will be bisected (*start* and *end*). *Start* is a hazard rate whose $F_R(t)$ does not meet the main requirement, *end* is a hazard rate whose $F_R(t)$ meets the main requirement.
   b) Perform the bisection until the required accuracy given by minStep is met.
   c) Verify the result.

The flowchart shown in Fig. 2 contains subroutines *verify value* and *find end*. *Verify value* checks whether $F_R(t)$ using selected *value* as hazard rate is greater than $F_E(t)$ all the time or not. In other words, it checks whether the *value* meets the main requirement.
The *find end* method is used to find the end of the interval that will be bisected. A new function $M(\lambda)$ is created as the best-fitting function using three points – three maximal differences between $F_E(t)$ and $F_R(t)$ using three consecutive hazard rates with minStep difference. The *end* value is taken as the hazard rate, where the extrapolation of function $M(\lambda)$ reaches 0.

IV. Partial Coverage Reduction

The main requirement (the reduced failure distribution function must be greater than the failure distribution function of the exact model all the time) mentioned in Section IV may be too strict in many applications.

The area, where the main requirement is not met, is mostly located in the upper part of the failure distribution function, where the probability of safety violation is close to 1. The safety-critical system has to be replaced/repaired long before this area is reached.

If the specifications/standards define that the system has to be replaced/repaired when the probability of safety violation meets the specified level, the rest of the failure distribution function can be ignored (the main requirement does not need to be met in the ignored area). The ignoring of the rest of the failure distribution function can significantly improve the accuracy of the reduction in the most interesting area of the failure distribution function, where the probability of safety violation is close to 0.

The plot shown in Fig. 3 contains the illustrative example of the reduced failure distribution functions using partial and full coverage. The horizontal axis represents the time of operation

[1] Numeric method performed by NDSolve command of Mathematica 8.01 [7] software is used in this paper.

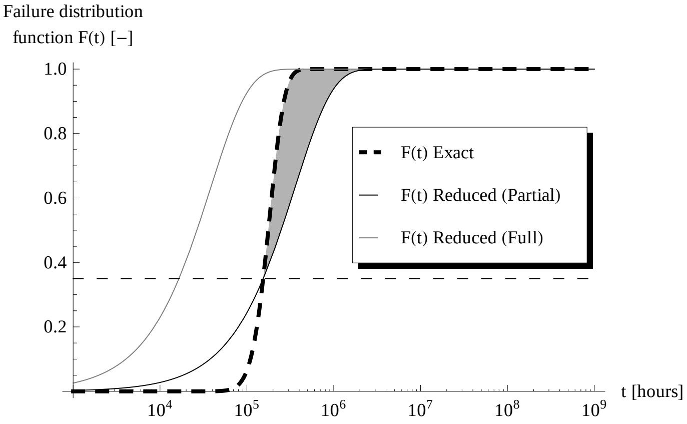

Fig. 3. Illustrative example of partial coverage of failure distribution function.

measured in hours, the vertical axis represents the failure distribution function. The thick dashed line represents the exact failure distribution function, the black line represents the reduced failure distribution function using the partial coverage, the gray line represents the reduced failure distribution function using the full coverage, and the horizontal line represents the limit value. The area, where the exact failure distribution function is greater than the reduced failure distribution function, is highlighted by a light-gray shading. As you can see, the area is located above the limit value, thus it is not taken into account during the reduction using the partial coverage.

V. Case Study System

A. Modified Duplex System Block

Modified Duplex System (MDS) is based on two independent modules with parity checkers attached [8]. The parity checkers are able to detect some faults. The rest of the faults are detected by comparators attached to the outputs of both blocks. The MDS is designed to utilize the reconfiguration ability of FPGA. The reconfiguration is able to repair a part affected by a fault in tenth of milliseconds.

The dependability model of MDS used in this paper is constructed using the following assumptions:

- Two faults will never occur in the block at the same time.
- When a fault occurs in one module, the parity checker attached to this module is able to detect the fault. If the fault is detected by parity checker, the affected module is repaired. If the fault is not detected by parity checker (it is detected by comparators), both modules have to be repaired, because the faulty module cannot be identified.
- 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 is used to calculate the exact failure distribution function $F_E(t)$ of the MDS block.

$Fault\_Free$ is the functional/fault-free state of the block. The fault rate of the first fault is $2\,\lambda$, because the first fault can affect two functional parts of the block. $Latent$ state is active when the block contains a fault that has not been detected yet.

The self-test rate is labeled as δ. If the self-test is performed successfully (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 the self-test is performed successfully (a fault is detected by the set of the 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.

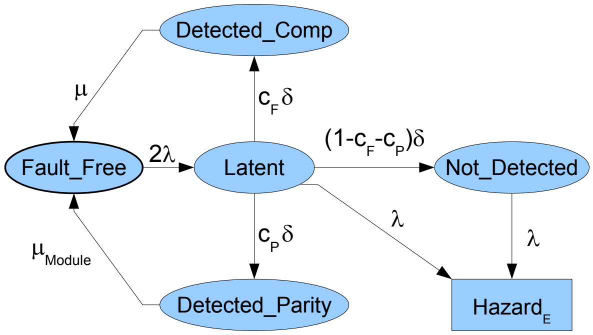

Fig. 4. Dependability model of Modified duplex system block used to calculate the exact failure distribution function.

If the self-test fails (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 part will lead to safety violation ($Hazard_E$ state). The second fault hit inside already affected functional part cannot cause a hazard, because the second functional part works correctly.

The arc leading from $Latent$ to $Hazard_E$ expresses the possibility that a second fault affects the unaffected functional part before the self-test is finished.

The block locked in the $Detected\_Parity$ state waits until the repair is finished (repair rate μ_{Module} – only one part is repaired). The block locked in the $Detected\_Comp$ state also waits until the repair is finished (repair rate μ – both parts and the set of the comparators is 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.

$\mu = 10^3\,[h^{-1}]$ – the repair rate of the whole block (both parts and the set of comparators)
$\mu = 5 \times 10^3\,[h^{-1}]$ – the repair rate of the faulty part
$\lambda = 10^{-5}\,[h^{-1}]$ – the fault rate
$\delta = 10^{-1}\,[h^{-1}]$ – the self-test 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 lower part of the illustrative example in Fig. 1.

The steps of reduction correspond to the algorithm described in Section IV.

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

The calculation is made using the system of differential equations.

Find an estimated value λ_{Hazard_Est}.

We use 10 different probability levels covering the interval

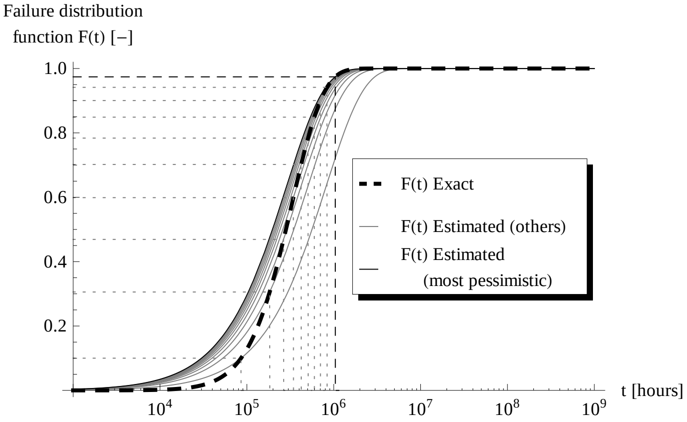

Fig. 5. Failure distribution function of the exact model ($F_E(t)$) and estimated functions intersecting $F_E(t)$ at predefined probability levels (MDS block).

$(0, 1)$ to gain more precise estimation in this paper. The levels are given by

$$level_i = 1.1 - 10^{-(\frac{i}{10})}, \text{ where } i = 0 \ldots 9$$

This non-linear level distribution is based on 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 the upper part of the interval $(0, 1)$ in most cases.

All 10 probability levels and estimations are shown in Fig. 5. The horizontal axis represents the time of operation measured in hours, the vertical axis represents the failure distribution function. The thick dashed line represents the exact failure distribution function, the black thin line represents the most pessimistic estimated failure distribution function used in the correction step, and the gray lines represent the other estimated failure distribution functions. The horizontal and vertical lines show the intersections of the exact and the estimated failure distribution functions.

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

$$\lambda_{Hazard\_Est} = 3.511 \times 10^{-6} \, [h^{-1}]$$

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 IV, because there is an area, where the exact failure distribution function is greater than the reduced failure distribution function. The correction made according to the flowchart (see Fig. 2 in Section IV) is necessary in such case.

The plot shown in Fig. 6 contains the exact failure distribution function, the estimated failure distribution function from the previous step and the reduced failure distribution function. The horizontal axis represents the time of operation measured in hours, the vertical axis represents the failure distribution function. The thick dashed line represents the exact failure distribution function, the gray line represents the estimated failure distribution function, and the black line represents the reduced (corrected) failure distribution function. The area, where the exact failure distribution function is greater than the reduced failure distribution function, is highlighted by a light-gray shading (see the zoom window).

The corrected value λ_{Hazard_MDS} is

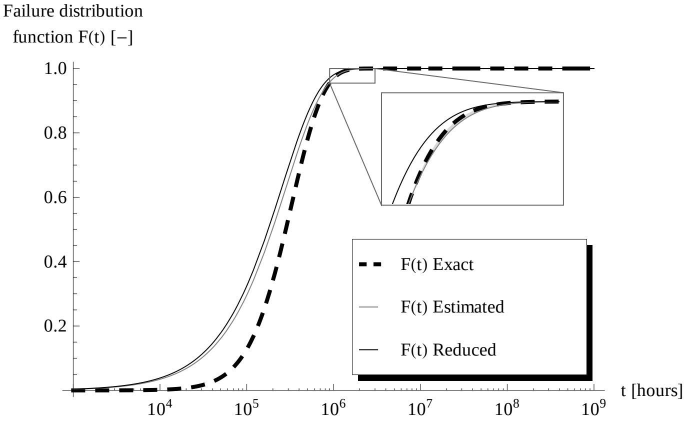

Fig. 6. Exact, estimated, and reduced failure distribution functions of the MDS block.

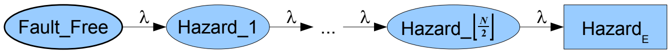

Fig. 7. Dependability model of generic N-modular redundant system used to calculate exact failure distribution function.

$$\lambda_{Hazard\_MDS} = 3.927 \times 10^{-6} \, [h^{-1}]$$

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

$$t_{Reduction\_MDS} = 12.48 \, [s]$$

This CPU-time will be used in Section V-D1 to compare runtimes of reduction of hierarchical and non-hierarchical dependability models of the system using MDS as a block.

B. *N-modular Redundancy*

The model shown in Fig. 7 is used to calculate the exact failure distribution function of a generic NMR system. The NMR system containing N blocks will contain $\lfloor \frac{N}{2} \rfloor$ transient hazard states. These states correspond to the numbers of blocks that are in the hazard state. We reduce NMR systems consisting of 1 (a single block) to 17 blocks in this section.

C. *NMR based on Modified Duplex System Blocks*

The hierarchical dependability model of the NMR system based on MDS blocks is shown in Fig. 8. The model of a MDS block is created, reduced, and the result of reduction (λ_{Hazard_MDS} calculated in Section V-A) is taken as the hazard rate (λ) of the NMR model. The result of the reduction of the hierarchical dependability model (λ_{Hazard}) is calculated in Section V-D.

D. *Results*

1) Comparison of Runtimes: Table I shows the comparison. The first time is the CPU-time spent on solving[3] the system of differential equations of the exact dependability model – model generated by the Cartesian product of the dependability models of the MDS blocks configured as NMR. The second time is the summarized CPU-time spent on reducing the MDS dependability model ($12.48\,s$ – $t_{Reduction\_MDS}$ taken from Section V-A) and CPU-time spent on reducing the dependability model of NMR. The reduction time includes the time required to solve the exact model, to estimate λ_{Hazard_Est} and to make the correction. Both models (MDS and NMR) are small, thus the main part of the time is spent on the corrections.

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

[3] NDSolve command of Mathematica 8.01 [7] software running on Intel Core i5 @3.3 GHz, OS: Win7 64-bit

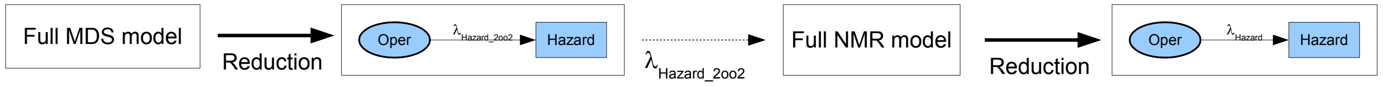

Fig. 8. Hierarchical dependability model of case study system.

TABLE I. NUMBER OF STATES AND CPU-TIMES OF SOLUTIONS OF N-MODULAR REDUNDANT SYSTEM BASED ON IDENTICAL MODIFIED DUPLEX SYSTEM BLOCKS.

NMR blocks	No. of states before merging	No. of states after merging	NDSolve time [s]	Reduction time [s]
n1 (MDS)[1]	6	6	0.031	12.48 [2]
n3	155	55	0.125	26.18
n5	3900	246	0.593	25.30
n7	97625	771	2.449	25.65
n9	2.441×10^{6}	1946	9.204	25.72
n11	61.03×10^{6}	4242	36.83	26.35
n13	1.526×10^{9}	8316	135.2	25.99
n15	38.15×10^{9}	15042	594.3	25.43
n17	953.7×10^{9}	25542	2041.3	25.40

[1] NMR containing one block is equivalent to a single MDS block.
[2] Contains time to reduce an MDS block only.

TABLE II. THE HAZARD RATES OF THE HIERARCHICAL MODEL USING PARTIAL COVERAGE REDUCTION.

Limit	$\lambda_{\mathbf{Hazard}}$	Ratio
10^{-4}	2.803×10^{-9}	9300
10^{-3}	20.30×10^{-9}	1284
10^{-2}	142.3×10^{-9}	183.2
0.1	959.0×10^{-9}	27.19
0.35	2.795×10^{-6}	12.19
0.6	4.762×10^{-6}	8.76
0.95	10.07×10^{-6}	2.59
0.99	12.75×10^{-6}	2.04
0.999	15.61×10^{-6}	1.67
1[1]	26.07×10^{-6}	1

[1] Limit 1 is equal to the reduction with the full coverage.

Some states of the exact Cartesian-product dependability model may be merged, because the MDS blocks are identical and it is not necessary to distinguish, which blocks are in a hazard state. The size of the model without merging grows too fast, so we generate the merged model – *Cartesian model* – directly.

The first column of Table I shows the number of the MDS blocks, the second column shows the number of the states of the Cartesian-product dependability model without state merging. The dependability model without state merging would be used, if the blocks were not identical. The third column shows the number of states of the Cartesian model (after state merging), the fourth column shows the CPU-time spent on solving the Cartesian model exactly and the fifth contains the sum of the CPU-times spent on reducing MDS ($t_{Reduction\_MDS}$) and reducing NMR.

As you can see, the CPU-time spent on solving the Cartesian model grows exponentially when the MDS blocks are added, but the reduction time is nearly constant, therefore the reduction will be faster when the system containing more than 11 blocks is used in this particular configuration.

2) Accuracy: Table II shows the hazard rates calculated using the hierarchical model. The first column shows the limit of probability of safety violation for main requirement application. The main requirement is applied up to this value only. The third column shows the ratio between the full coverage reduction and the partial coverage reduction with the given limit value.

As you can see, the lower the limit the higher hazard rate is able to meet the main requirement. The best ratio of the presented systems is cca. 10000, when the limit is 10^{-4}.

VI. CONCLUSIONS

A method of constructing of hierarchical dependability models based on Markov chains has been presented. The hierarchical models can be used to calculate the hazard rate – the rate of a hazard event leading to a situation where safety of a system is violated. The hazard rate is the key value specifying whether the hazard event may be tolerated or not.

The presented models are applicable to many mission-critical and safety-critical systems including railway systems, automotive systems, etc.

The proposed hierarchical model has been used to calculate the hazard rate of a complex system. The system uses two-level redundancy – Modified duplex system method as low-level redundancy and N-modular as high-level redundancy. The results indicate that the hazard rate calculated using hierarchical dependability model based on Markov chains is higher than the hazard rate calculated without using hierarchy, but the increase can be significantly reduced, when the partial coverage of the failure distribution function is applied.

The CPU-time spent on the reduction of the hierarchical dependability model based on Markov chains is greatly reduced (up to 80 times) compared to the same system modeled by a non-hierarchical model. The partial coverage of the failure distribution function could lead to additional acceleration of the reduction, but the partial coverage reduction algorithm.

The results also show that the CPU-time spent on solving the system of the differential equations of the dependability model generated by the Cartesian product of the dependability models of the blocks grows exponentially with respect to the number of blocks used, but the CPU-time spent on solving the system of the differential equations of the hierarchical dependability model is nearly constant.

ACKNOWLEDGMENT

This research has been partially supported by the projects SGS12/094/OHK3/1T/18 and SGS13/017/OHK3/1T/18.

REFERENCES

[1] Normand, E.: Single Event Upset at Ground Level, IEEE Transactions on Nuclear Science, vol. 43, 1996, pp. 2742–2750.

[2] Avižienis, A., Laprie, J.-C., Randell, B. and Landwehr C.: Basic Concepts and Taxonomy of Dependable and Secure Computing, IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 1, January–March 2004.

[3] Pradhan, D., K.: Fault-Tolerant Computer System Design, Prentice Hall PTR, Upper Saddle River, New Jersey 1996, ISBN 0-7923-7991-8.

[4] Hoang, P.: System Software Reliability, Chapter 2: System Reliability Concepts, Springer Series in Reliability Engineering, Springer London (2007), pp. 9–75.

[5] Electronic Reliability Design Handbook – MIL-HDBK-338. Web: https://assist.daps.dla.mil/quicksearch/basic_profile.cfm?ident_number=54022

[6] Sahner, R., A.; Trivedi, K., S.: Reliability Modeling Using SHARPE, IEEE Transactions on Reliability, vol. R-36 (1987), pp. 186–193.

[7] Wolfram Mathematica web page. http://www.wolfram.com/mathematica/

[8] Kubalík, P. and Kubátová, H.: Dependable design technique for system-on-chip, Journal of Systems Architecture, Vol. 2008, no. 54, (2008) 452–464. ISSN 1383-7621.