Non-Homogeneous Continuous Time Markov Chains Calculations

Jan Řezníček, Martin Kohlík, Hana Kubátová
Department of Digital Design, Faculty of Information Technology
Czech Technical University in Prague, Technická 9
Prague, Czech Republic
{reznija5, kohlimar, hana.kubatova}@fit.cvut.cz

***Abstract*—Dependability models allow calculating the rate of events leading to a hazard state – a situation, where safety of the modeled dependable system is violated, thus the system may cause material loss, serious injuries or casualties. This paper shows a method of calculating the hazard rate of the non-homogeneous Markov chains using different sets of homogeneous differential equations for several hundreds small time intervals (using default parameters settings – the number of the intervals can be adjusted to balance accuracy/time-consumption ratio). The method is compared to a previous version based on probability matrices and used to calculate the hazard rate of the hierarchical Markov chain. The hierarchical Markov chain allows us to calculate the hazard rates of the blocks independently and the non-homogeneous approach allows us to use them to calculate the hazard rate of the whole system. This method will allow us to calculate the hazard rate of the non-homogeneous Markov chain very accurately compared to methods based on homogeneous Markov chains.**

***Index Terms*—Fault tolerant systems, Hierarchical systems, Reliability, Reliability engineering.**

I. Introduction

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

The presented method is used to compute failure distribution function of the time-continuous Markov chains. The previous method presented in [1], [2] is focused on computations of state-based dependability models by probability matrix multiplications with non-homogeneous failure parameters. The method is used to compute complex systems by creating hierarchical model and we use the method repeatedly for each part of the model. The disadvantage of this method is complication that we must transfer the hazard rate matrix into probability matrix in every step to use this method.

The method proposed in this paper is not based on matrix multiplication, instead we use differential equations to compute the failure distribution function $F(t)$.

The proposed method is demonstrated on a case study containing multiple (up to 9) 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 case study is used to calculate the total hazard rate of the system and to demonstrate the dependencies of time-consumption and accuracy on the parameters of the proposed method. The models we use in this paper are same as in [1], [2] to compare both methods of computing.

The paper is organized as follows: Section II provides the theoretical background and introduces Markov chains. Section III describes the proposed method. The results are shown in Section IV and Section V concludes the paper.

II. Theoretical Background

The proposed method is used to calculate the samples of failure distribution function using non-homogeneous continuous Markov chains, thus both homogeneous and non-homogeneous Markov chains are introduced in this section. The first part of this section also contains description of differences between computing Markov chains by matrix multiplications and solving differential equations. The calculation of failure distribution function and its discrete samples is summarized in the second part of this section.

A. The Continuous Time Markov Chains

The continuous time Markov chains are defined by the hazard rate matrix Q. The matrix Q have elements q_{ij} that are describing the intensity of transition from state i to state j. The elements q_{ii} are defined as a complement of the sum of other elements in the row i (the sum of all elements in the row i is 0).

We have two ways to work with hazard rate matrix Q:

- **Matrix Multiplication** – This method described in [1], [2] is based on converting the matrix Q into P by the equation:
$$P = I + \frac{1}{\Delta}Q \tag{1}$$
where Q is the hazard rate matrix, P is final probability matrix, I is identical matrix and Δ is the highest value of diagonal elements q_{ii} in matrix Q.
- **Differential equations** – The hazard rate matrix Q can be converted to the system of differential equations:
$$p_i'(t) = -q_{ii} * p_i(t) + \sum^{j} q_{ji} * p_j(t) \tag{2}$$

978-1-7281-9535-3/20/$31.00 ©2020 IEEE
DOI 10.1109/DSD51259.2020.00108

where $p'_i(t)$ is probability to transition to state i in time t, q_{ii} is the hazard rate in matrix Q on diagonal, that marks the hazard rate of transition from state i, q_{ji} is the hazard rate of transition from state j to state i and $p_j(t)$ is probability of being in the state j in time t. For this equation hold that $i, j \in 0, \ldots, n; i \neq j$, where n is number of state in Markov chain.

B. Markov Chains and Cumulative Failure Distribution Function

An *absorbing* state i is a state that, once entered, cannot be left. Therefore, the state i is absorbing if the following condition is met:

$$p_{ii} = 1 \text{ and } p_{ij} = 0 \text{ for } i \neq j \tag{3}$$

The evolution of the probability distribution of the absorbing states over time forms the series of the samples of the *cumulative (failure) distribution function* $F(t)$ defined as the probability in a random trial that the random variable is not greater than t, or

$$F(t) = \int_{-\infty}^{t} f(t)\, dt \tag{4}$$

C. Non-Homogeneous Markov Chains

Both discrete time Markov chains and continuous time Markov chains can be defined using variable matrix elements. Such systems are called non-homogeneous Markov Chains.

Most methods used to calculate the probability distribution of states and their evolution of over time cannot be used in this case, but there are methods intended for non-homogeneous Markov chains.

There are many methods based on estimation (e.g. maximum likelihood) of unknown parameters applied to models from a wide range of scientific disciplines, e.g. smart-city [3], ecology [4], medicine [5], and others [6], but their main goal is similar to the methods referenced in the previous paragraph – to estimate the unknown parameters (transition rates) of an Markov chain.

Another method presented in [7] splits a single transition to several stages (intermediate states and transitions). All intermediate transitions are exponentially distributed, but their concatenation can emulate the behavior of another (e.g. Erlang) distribution. There are two main disadvantages of this method: new states have to be added to the Markov chain and the method can emulate behavior of a limited set of distributions only.

The principle similar to our method – dividing the time interval, where the evolution of the probability distribution of states is calculated, to several "slices" – is presented in [8]–[10]. Each time-slice has a constant probability/transition matrix, ie. the Markov chain is homogeneous in this time-slice. There are two main differences between these and our method:

1) The number of the time-slices – we use more (several hundreds by default) time-slices, thus the accuracy is significantly improved and the accuracy/time-consumption ratio can be controlled using method parameters.
2) The main goal – the referenced methods are used to estimate the unknown parameters (transition rates) of an Markov chain, our method takes an Markov chain with known parameters and calculate the evolution of the probability distribution of the states over time and the failure rate of the Markov chain.

III. Proposed Method Description

The presented method has the same two parts of evaluating as the method from [2]:

1) Calculate the failure distribution function of non-homogeneous model.
2) Calculate the hazard rate for each time-slice (interval between two consequential samples) calculated in the previous part.

A. Calculation of the Failure Distribution Function of Non-homogeneous Model

The flowchart of calculation of this method is shown in Figure 1.

1) The first step of the computation is initializing the differential equations and theirs initial conditions. These conditions are defined in time $t = 0$. Next step of the initialization is solving the differential equations by the NDSolve method in Wolfram Mathematica [11] for time $t \in [0; t_{start}]$, where $t_{start} = 2^{expStart}$ is starting value for our results of this method. Than we set the initial conditions of the Markov chain to values for time $t = t_{start} = 2^{expStart}$. The method create initial time variable $t_{current}$ that is used as the main time entry for computation cycle and also to finish calculating the values for failure distribution function $F(t)$. The method already create time-shift variable Δt_{sample} that works as rising coefficient for variable $t_{current\_homo\_end}$.
2) At the beginning of each cycle the method set new $t_{current\_homo\_end} = t_{current\_homo\_end} + \Delta t_{sample}$ and get new values of hazard rate parameters to $params_{new}$.
3) In next step the method check the value of $t_{current\_homo\_end} = 2^x, x \in 1, 2, \ldots$. In positive situation the time-shifting coefficient is doubled ($\Delta t_{sample} = 2 * \Delta t_{sample}$).
4) In next part of the computation the method compare current hazard rate parameters in variable $params_{new}$ with last using value of the parameters in variable $params_{current}$ and the final time $t_{current\_homo\_end} \geq t_{finish}$. The method has two options:
   a) The time $t_{current\_homo\_end}$ is not at the end and values $params_{new}$ and $params_{current}$ are **equal** – the calculation continues to step 2),
   b) The values $params_{new}$ and $params_{current}$ are **not equal** – the method use the NDSolve function for differential equations with initial conditions $p_i(t_{current})$ and parameters in $params_{current}$ for time $[t_{current}, t_{current\_homo\_end}]$. After that

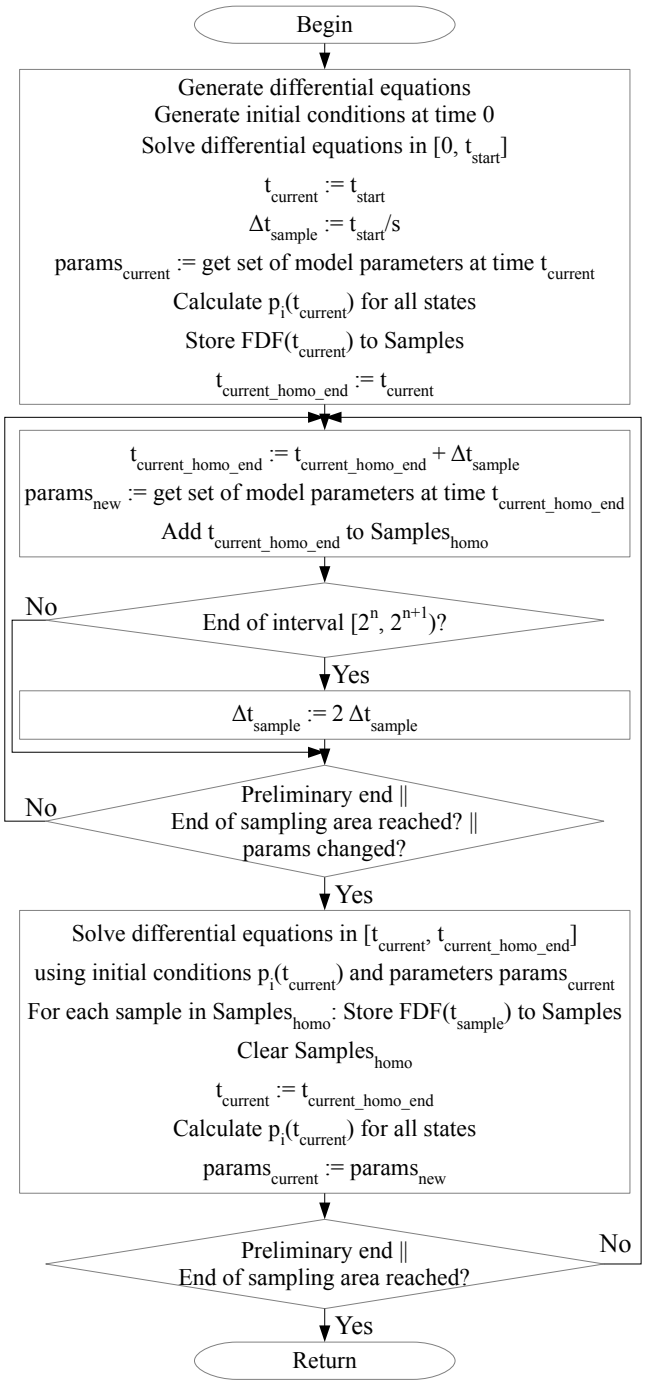

Fig. 1. Flowchart of calculation of failure distribution function of non-homogeneous model.

the method gets required values of failure distribution function $F(t)$ from NDSolve by setting the values retrospectively as the $t_{current\_homo\_end}$ and Δt_{sample} was. After that the method set the variables $t_{current} = t_{current\_homo\_end}$, $params_{current} = params_{new}$ and calculates new initialization conditions $p_i(t_{current})$ for all states. Finally the method goes to step 2),

c) The time $t_{current\_homo\_end}$ is at the end – the method use the same functionalities as in step 4b) but at the end it goes to step 5).

5) The result of this method is the list of the failure distribution function $F(t)$ values for time $t \in [t_{start}; t_{finish}]$.

B. Calculate the Hazard Rate for Each Time-slice

We use the same equation to computing the failure rate λ as in [1], [2]:

$$\lambda_i = \frac{\log_e(1 - F(t_i)) - \log_e(1 - F(t_{i+1}))}{t_{i+1} - t_i} \tag{5}$$

We also use same methods as we use in mentioned papers. We can compute the failure distribution function $F(t)$ universal in both methods – the matrix multiplication and presented method.

C. Parameters using in presented method

In Figure 2 we can se all parameters we defined as parameters of the presented method:

- s – parameter to set the number of using values in time $t \in [2^x; 2^{x+1})$,
- $epsilon$ – parameter to set the maximal value of the failure distribution function $F(t)$ in the computation,
- t_{start} – this parameter sets the starting time t for the method. The method set initialization until this time t and than start the main cycle from this time,
- t_{end} – this parameter sets the final time t for the method. In main cycle if we reach this time, the method is finalizing the results and end.

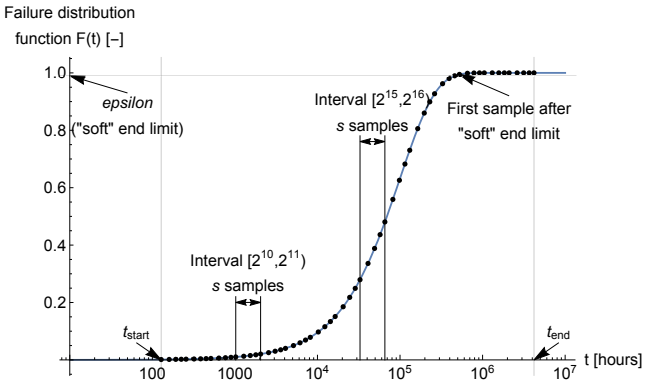

Fig. 2. Dependability model of Two-out-of-two block.

We can use similar parameter to filter values at the start of the method such as parameter $epsilon$ at the end the method.

The parameter $epsilon$ is defined that the value from failure distribution function $F(t)$, that are less then $1 - epsilon$, are important for the method. We can define parameter xi and every value from failure distribution function $F(t)$ that is bigger than this xi value, are important for our method.

In this paper, we use kind of this parameter xi in Section IV to filtered results of failure distribution function $F(t)$ that are less then $xi = 10^{-6}$. These values are very small and in relative errors comparison create significant differences.

IV. Results

A. Case Study Description

The proposed method is demonstrated on a case study containing multiple (up to 9) 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 case study is used to demonstrate the dependencies of time-consumption and accuracy on the parameters of the proposed method. Wolfram Mathematica [11] tool is used to perform the calculations. The models using for testing are same as in [1], [2] to compare both methods.

The dependable blocks used in the case study system use Two-out-of-two (2oo2) redundancy [12], [13]. 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 detailed description of the system and its model follows and can be found in [1], [2].

The model shown in Figure 3 is used to calculate the failure distribution function $F(t)$ of the 2oo2 block.

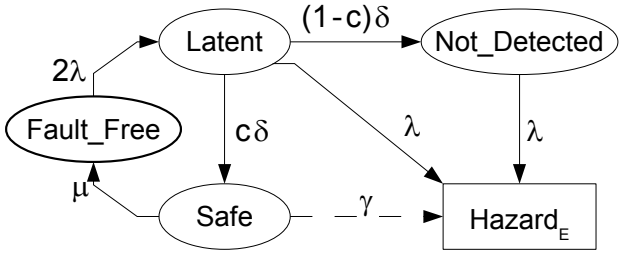

Fig. 3. Dependability model of Two-out-of-two block.

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 [13].

$\mu = 24^{-1}\,[hours^{-1}]$ – the repair rate
$\lambda = 10^{-5}\,[hours^{-1}]$ – the fault rate
$\delta = 10^{-1}\,[hours^{-1}]$ – the block-lock rate
$c = 0.6$ – the probability of detecting a fault by the block-lock
$\gamma = 10^{-3}\,[hours^{-1}]$ – the backup/emergency method hazard rate

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.

The model shown in Figure 4 is used to calculate the failure distribution function of a generic NMR system. The NMR system containing N blocks will contain $\left\lfloor \frac{N}{2} \right\rfloor$ transient states. These states correspond to the blocks that are in the hazard state. NMR systems consisting of 3 to 9 blocks are used in this paper.

B. Method Parameters Impact

The 3-MR (TMR) system based on 2oo2 blocks is used to demonstrate the impact of method s parameters. The model of this system is made as the Cartesian product of the dependability models of three identical 2oo2 blocks and the model of the TMR. The model contains 34 states.

For testing presented method we use only the parameter s, that is the same in matrix multiplication method [1], [2].

The other parameters values are as follows:

$t_{start} = 1\,[hour]$ – time, where the first sample of the failure distribution function $F(t)$ is stored
$t_{end} = 2^{20} \doteq 10^6\,[hours]$ – time of the "hard" end
$SoftEndEnabled =$ false

Table I shows the impact of the *s* parameter on both methods – matrix multiplication method in [1], [2] and presented method.[1]

The size of *s* is shown in the first column, the *Initialization* CPU-time[2] spent on is shown in the second and fourth column. The third and fifth column shows *Main loop* CPU-time.

The initialization time of matrix multiplication method and the differential equations method in Table I are not same times. The initialization time in matrix multiplication method is computing by calling the method `CreateInitMatrix`, that is described in [1], [2]. In presented method the initialization time only compute the first initialization of differential equations between time $t = 0$ and $t = t_{start}$. In our testing we set the parameter t_{start} to value 1.

C. Time Non-Homogeneity Impact

The values in Table I are based on the fact, that the testing model is homogeneous (all the fault parameters are constant in time t). In this section we will show the time-computing and accuracy differences between the homogeneous and non-homogeneous type of presented method. The body of this method has no differences, but in every time-slice we shut down the control condition

[1]The method presented in [2] contained a bug: The interval could be doubled at the beginning of the failure distribution function $F(t)$ ($t_{current} == t_{start}$). This bug cut the number of the time-slices (s parameter) by half. All results presented in this paper have been corrected – all time and error values are kept, but the values of the s parameter have been lowered.

[2]Running on Intel Core i5-7300HQ @2.5 GHz, OS: Win10 64-bit, Mathematica 12.1.

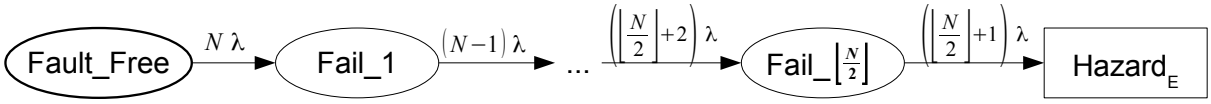

Fig. 4. Dependability model of generic N-modular redundant system.

TABLE I
COMPARISON OF CPU-TIMES OF THE METHOD WITH RESPECT TO THE *s* PARAMETER.

	Matrix Multiplication		Differential Equations	
s [-]	**Init. time [s]**	**Main loop time [s]**	**Init. time [s]**	**Main loop time [s]**
2^1	0.130	0.210	0.031	0.516
2^2	0.130	0.330	0.031	0.531
2^3	0.130	0.560	0.031	0.656
2^4 [1)]	**0.130**	**0.962**	**0.031**	**0.875**
2^5	0.125	1.840	0.031	1.266
		...		
2^{10}	0.126	57.40	0.031	32.81

[1)] Default value used in [2] and this paper.

$params(t_{current}) \neq params(t_{last})$. By canceling this condition the final part of the method in Figure 1 is processing in each time slice.

We can compare the accuracy differences between matrix multiplication method and the presented method only in Section IV-D. There is a comparison between MMm, DEm and the analytical solution of the NMR blocks. In other sections we can compare the computing-time of both methods. The accuracy differences are compared only in differential equations method. We can not compare both methods in other sections because the missing analytic solution for NMR system of two-out-of-two blocks.

Table II shows comparison of computing speed of presented method and matrix multiplication method for TMR (3-MR) Markov chain based on three two-out-of-two blocks. In first column we see the type of system. In second column we see the time to evaluate the matrix multiplication method. The time of differential equations method is in the last column.

TABLE II
COMPARISON OF CPU-TIMES OF HOMOGENEOUS AND NON-HOMOGENEOUS CASE.

Case type	**Matrix Multiplication Method time [s]**	**Differential Equations Method time [s]**
Homogeneous	1.160	0.920
Non-Homogeneous	17.97	11.25

In Table III we see relative errors of the non-homogeneous solution and the homogeneous option. In first column we see number of two-out-of-two blocks in the system, in second column we see the worst relative error of the failure distribution function $F(t)$ of non-homogeneous option from homogeneous model. The average relative error of this function we see in the last column.

TABLE III
COMPARISON OF RELATIVE ERRORS OF HOMOGENEOUS AND NON-HOMOGENEOUS CASE WITHOUT USING FILTER xi.

NMR blocks	**Worst rel. error [-]**	**Average rel. error [-]**
3-MR	2.49×10^{-2}	1.52×10^{-3}
5-MR	1.28×10^{-1}	7.28×10^{-3}
7-MR	2.67×10^{-1}	1.36×10^{-2}
9-MR	3.28	4.70×10^{-2}

The plot shown in Figure 5 shows the relative errors of all samples between the homogeneous and the non-homogeneous case. The horizontal axis represents the time of operation measured in hours, the vertical axis represents the size of the relative error.

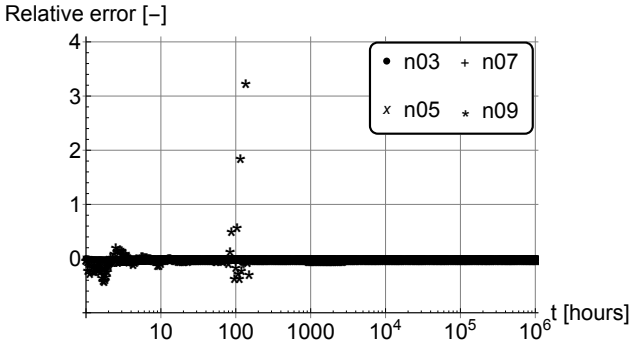

Fig. 5. Relative errors of all samples between homogeneous and non-homogeneous case without using filter xi.

As we see in the Figure 5, the highest peaks of differences in the results are in time-slices that the failure distribution function $F(t)$ value is less then 10^{-6}. In this case are the relative errors very big even when the absolute error can be only in 10^{-7}.

The Table IV is close to the Table III, but we filtered the values of failure distribution function $F(t)$ that are less the parameter xi, that has value $xi = 10^{-6}$.

The plot shown in Figure 6 shows the relative errors of all samples between the homogeneous and the non-homogeneous case. The horizontal axis represents the time of operation measured in hours, the vertical axis represents the size of the relative error. In this plot we use the filtered data (by filtering values that are lower than $xi = 10^{-6}$).

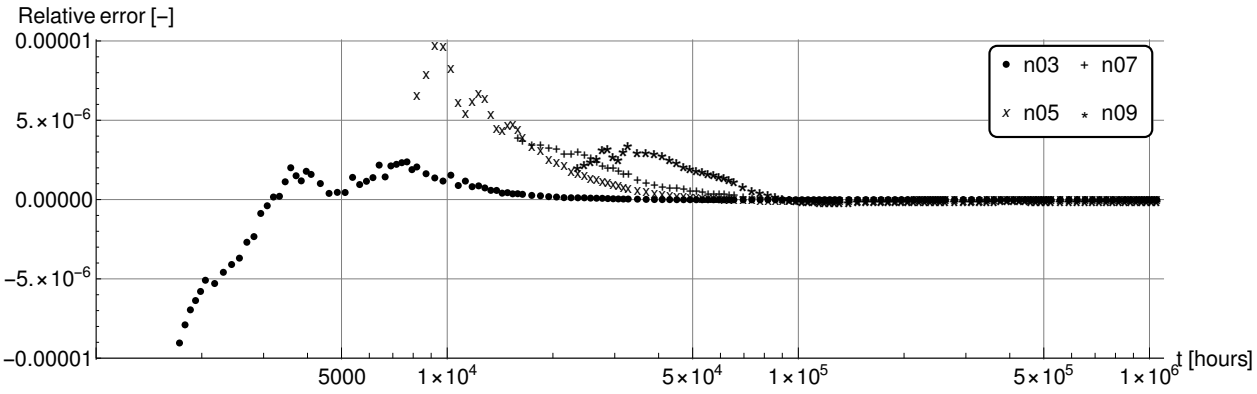

Fig. 6. Relative errors of all samples between homogeneous and non-homogeneous case using filter xi.

TABLE IV
COMPARISON OF RELATIVE ERRORS OF HOMOGENEOUS AND NON-HOMOGENEOUS CASE WITH USING FILTER xi.

NMR blocks	Worst rel. error [-]	Average rel. error [-]
3-MR	9.04×10^{-6}	7.59×10^{-7}
5-MR	9.71×10^{-6}	1.20×10^{-6}
7-MR	3.94×10^{-6}	6.29×10^{-7}
9-MR	3.52×10^{-6}	7.41×10^{-7}

TABLE V
COMPARISON OF RELATIVE ERRORS OF PRESENTED METHOD, MATRIX MULTIPLICATION METHOD AND ANALYTICAL SOLUTION.

	Matrix Multiplication		Differential Equations	
NMR blocks	Rel. error of the first sample [-]	Average rel. error [-]	Worst rel. error [-]	Average rel. error [-]
3-MR	-2.98×10^{-8}	1.69×10^{-8}	4.49×10^{-6}	3.47×10^{-7}
5-MR	-8.94×10^{-8}	4.92×10^{-8}	2.81×10^{-4}	1.67×10^{-5}
7-MR	-1.79×10^{-7}	9.76×10^{-8}	5.41×10^{-5}	2.55×10^{-6}
9-MR	-2.98×10^{-7}	1.62×10^{-7}	7.00×10^{-5}	2.81×10^{-6}

D. Comparison to Analytical Solution

The direct comparison of accuracy for both methods is comparing to analytical solution. We compare the NMR model by the presented method, the matrix multiplication model and the analytical solution in this section. The failure distribution function of the NMR system can be calculated using the following equation

$$F_{NMR}(t) = 1 - \sum_{i=M}^{N} \binom{N}{i} F(t)^{N-i}(1 - F(t))^{i} \qquad (6)$$

where $F(t)$ is the failure distribution function of the single block ($F(t) = 1 - e^{(-\lambda \cdot t)}$ and $\lambda = 10^{-5}$ in this case), N is (odd) number of blocks used in the system, and M is number of blocks required to be operational ($\frac{N+1}{2}$ in this case).

Table V shows the relative errors between the presented method, Matrix Multiplication method from [1], [2] and the analytical solution. The first column shows the number of the NMR blocks, the second column shows the relative error of the first sample of the matrix multiplication method (compared to the sample taken from the analytical solution). The fourth column shows the worst relative error of the presented method. The average of the absolute values of relative errors of all stored samples is shown in the third and last column.

In Table V in presented method we use filtered data that are higher than $xi = 10^{-6}$. In case that we don't use this filter, the matrix multiplication method will be distinctly better.

The plot shown in Figure 7 shows the relative errors of the samples between the presented method and the analytical solution. The horizontal axis represents the time of operation measured in hours, the vertical axis represents the size of the relative error. Please note that only each third sample is shown – the plot would be hard to read when all samples were present.

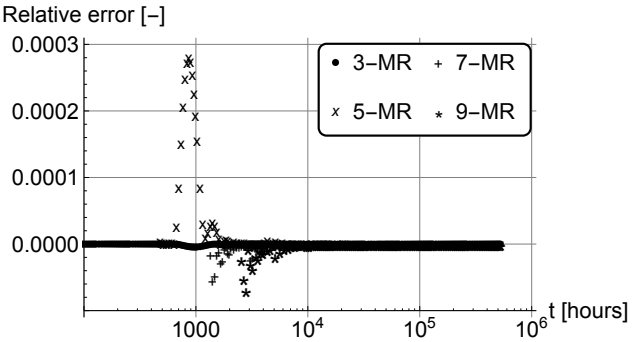

Fig. 7. Relative errors of samples between presented method and Analytical solution using filter by the parameter xi.

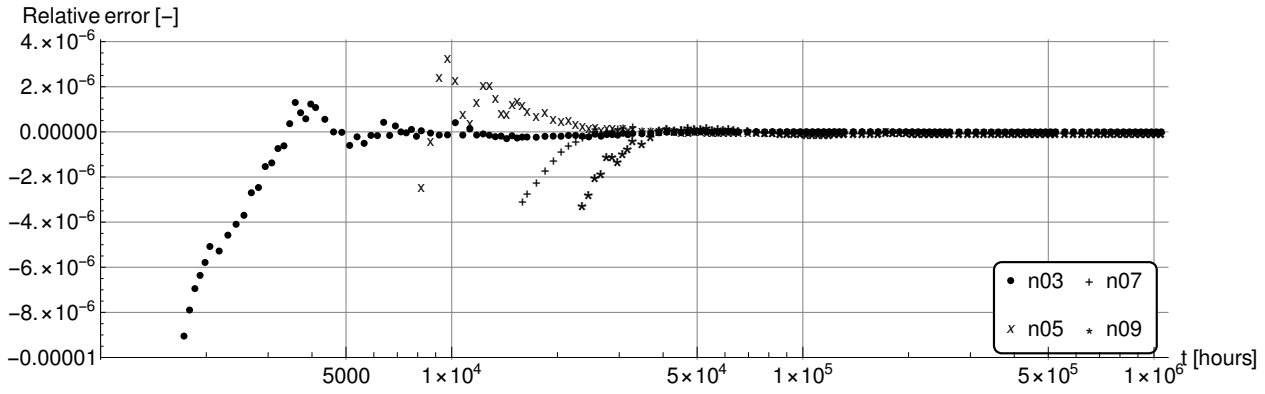

Fig. 8. Relative errors of samples between exact and hierarchical case using filtered data by parameter xi.

E. Difference between Cartesian-product and Hierarchical Approach

The hierarchical dependability model of the NMR system based on 2oo2 blocks is used in this case. The model of a 2oo2 block is created, the samples of the $F(t)$ function are calculated, and the hazard (failure) rate result is taken as the fault rate (λ) of the NMR model.

The results of the hierarchical approach are compared to the results of the exact model (the model generated by the Cartesian product of the dependability models of the 2oo2 blocks and the model of the NMR).

Table VI shows the comparison of the CPU-times and the relative errors of the hierarchical and the exact solutions. The first column shows the number of the 2oo2 blocks, the second column shows the CPU-time spent on exact model solution (sum of *Initialization* and *Main loop* CPU-times). The CPU-time spent on hierarchical method (*Initialization* and *Main loop* CPU-times for the 2oo2 model and the same times spent on the NMR model) is shown in the third column. The relative error of the first sample (compared to the sample taken from the exact case) is shown in the third column. The average of the absolute values of relative errors of all stored samples is shown in the last column.

TABLE VI
COMPARISON OF RELATIVE ERRORS OF EXACT AND HIERARCHICAL CASE USING FILTERED DATA BY THE PARAMETER xi.

NMR blocks	Exact time [s]	Hierarchical time [s]	The Worst rel. error [-]	Average rel. error [-]
n03	0.922	1.422	9.05×10^{-6}	5.54×10^{-7}
n05	4.343	1.610	3.26×10^{-6}	2.71×10^{-7}
n07	20.44	1.828	3.07×10^{-6}	1.61×10^{-7}
n09	102.0	1.968	3.18×10^{-6}	2.12×10^{-7}

The plot shown in Figure 8 shows the relative errors of the samples between the exact and the hierarchical case. The horizontal axis represents the time of operation measured in hours, the vertical axis represents the size of the relative error. Please note that only each third sample is shown – the plot would be hard to read when all samples were present.

The CPU-time spent on solving the exact model solution grows rapidly with increasing number of the blocks, but the CPU-time spent on hierarchical method is below 1 second in all presented cases.

The maximal value of the relative error is slowly increasing with increasing number of the blocks, but it remains very low in all presented cases (ca. 10^{-7}).

For example, in Table VII we have computing-time of 9-MR block by the matrix multiplication method and the presented method.

TABLE VII
COMPARISON OF CPU-TIMES OF EXACT AND HIERARCHICAL CASE OF BOTH METHODS FOR 9-MR BLOCK.

Total Time	Exact Time [s]	Hierarchical Time [s]
Matrix Multiplication	5,938	0.225
Differential Equations	102.0	1.968

From Table VII the presented method have better computing-time for more complex exact models than the matrix multiplication method.

V. CONCLUSIONS

This paper is about new method for evaluating the Failure Distribution Function $F(t)$ from non-homogeneous Markov chains. This method is using differential equations for computing time-slices with set of adjustable parameters. The method can also compute with different values of Markov chain parameters in each time-slice. New method have a lot in common with the method presented in [1], [2], which was based by probability matrix multiplication instead the differential equations.

The presented method has similar calculation time to the matrix multiplication method in case of homogeneous model. In that case we use two intervals only, the values of failure distribution function $F(t)$ are find out at the end of calculation.

In case of non-homogeneous model the method is more time-consuming because each time-slice is calculated separately. Despite this time consuption this method has similar time results as the matrix multiplication method.

The accuracy of the calculation is worse if we compare the whole failure distribution function $F(t)$. The highest peaks of the errors are in the area, where the function value is less than $xi = 10^{-6}$. In the other part of this function, the relative errors are approx 10^{-7}.

The s parameter has influence to number of resulting values. If we want the failure distribution function $F(t)$ more accurate, we need bigger value of parameter s. Bigger value of s parameter also leads to longer evaluation time.

The proposed method is less accurate then the matrix multiplication method. It is using interpolation functions computing, so we can have all of function values over time interval t that we defined at the beginning. The matrix multiplication method computes only isolated points.

Method presented in this paper may be more preferable for evaluating bigger and more complicated systems. With rising number of states of Markov chain, the computing time in presented method has slight growth than matrix multiplication method. The computing time in hierarchical case has very good progress to reduce time to evaluate more complex systems.

ACKNOWLEDGMENT

This research has been partially supported by the OP VVV funded project CZ.02.1.01/0.0/0.0/16_019/0000765 “Research Center for Informatics”, and Czech Technical University grant SGS20/211/OHK3/3T/18.

REFERENCES

[1] J. Řezníček, M. Kohlík, and H. Kubátová, “Hierarchical Dependability Models based on Non-Homogeneous Continuous Time Markov Chains”, 14th International Conference on Design & Technology of Integrated Systems In Nanoscale Era (DTIS), Mykonos, Greece, 2019, pp. 1–2.

[2] J. Řezníček, M. Kohlík, and H. Kubátová, “Accurate Inexact Calculations of Non-Homogeneous Markov Chains”, 22nd Euromicro Conference on Digital System Design (DSD), Kallithea, Greece, 2019, pp. 470–477.

[3] E. B. Iversen, J. K. Møller, J. M. Morales, and H. Madsen, “Inhomogeneous Markov Models for Describing Driving Patterns”, In IEEE Transactions on Smart Grid, Vol. 8, Num. 2, pp. 581–588, March 2017.

[4] T. Scholz, V. Lopes, and A. Estanqueiro, “A cyclic time-dependent Markov process to model daily patterns in wind turbine power production”, Journal of Energy, Vol. 67, pp. 557–568, April 2014.

[5] S.-L. Pan, H.-H. Chen, “Time-varying Markov regression random-effect model with Bayesian estimation procedures: Application to dynamics of functional recovery in patients with stroke”, Journal of Mathematical Biosciences, Vol. 227, Iss. 1, pp. 72–79, 2010, ISSN 0025-5564

[6] D. Pouzo, Z. Psaradakis, and M. Sola, “Maximum Likelihood Estimation in Possibly Misspecified Dynamic Models with Time Inhomogeneous Markov Regimes”, SSRM Electronic journal, 2016. Accessed: 2020-04-28. Available from SSRN: https://ssrn.com/abstract=2887771

[7] V. Vais, S. Racek, “Experimental evaluation of regular events occurrence in continuous-time markov models”, In Proceedings of the Eleventh International Conference on Informatics, Košice, Slovakia, 2011, pp. 143–146.

[8] L.D. Sharples, G.I. Taylor, and M. Faddy, “A piecewise-homogeneous Markov chain process of lung transplantation”, Journal of Epidemiology and Biostatistics, Vol. 6, Iss. 4, pp. 349–355, 2001.

[9] R. Ocaña-Rilola, “Two Methods To Estimate Homogenous Markov Processes”, Journal of Modern Applied Statistical Methods: Vol. 1, Iss. 1, Article 17, 2002.

[10] R. Ocaña-Rilola, “Non-homogeneous Markov Processes for Biomedical Data Analysis”, Biometrical Journal: Vol. 47, Iss. 3, 2005.

[11] Wolfram. Mathematica. 2019. Accessed: 2020-04-11. Available from: http://www.wolfram.com/mathematica/

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

[13] R. Dobiáš, H. Kubátová, “FPGA based design of the railway’s interlocking equipments”, In *Digital System Design, 2004. DSD 2004. Euromicro Symposium on*, Aug 2004, pp. 467–473.