Proceedings of ICONE-23
23rd International Conference on Nuclear Engineering
May 17-21, 2015, Chiba, Japan

ICONE23-1590

METRIC-BASED APPROACH AND TOOL FOR MODELING THE I&C SYSTEM USING MARKOV CHAINS

Valentyna Butenko
National Aerospace University "KhAI"
Kharkiv, Ukraine
v.odarushchenko@csn.khai.edu

Vyacheslav Kharchenko
Centre of Safety Infrastructure Oriented Research and Analysis,
Kharkiv, Ukraine
v.kharchenko@csis.org.ua

Elena Odarushchenko
Poltava National Technical University
Poltava, Ukraine
skifs2007@mail.ru

Dmitriy Butenko
Custom Software Development Company AltexSoft
Kharkiv, Ukraine
dd.butenko@gmail.com

Keywords: Instrumentation and control system, reactor trip system, Markov chain, metric, stiffness, decomposability, sparsity, fragmentedness

ABSTRACT

Markov's chains (MC) are well-know and widely applied in dependability and performability analysis of safety-critical systems, because of the flexible representation of system components dependencies and synchronization. There are few roadblocks for greater application of the MC: accounting the additional system components increases the model state-space and complicates analysis; the non-numerically sophisticated user may find it difficult to decide between the variety of numerical methods to determine the most suitable and accurate for their application. Thus obtaining the high accurate and trusted modeling results becomes a nontrivial task.

In this paper, we present the metric-based approach for selection of the applicable solution approach, based on the analysis of MCs stiffness, decomposability, sparsity and fragmentedness. Using this selection procedure the modeler can provide the verification of earlier obtained results. The presented approach was implemented in utility MSMC, which supports the MC construction, metric-based analysis, recommendations shaping and model solution. The model can be exported to the well-known off-the-shelf mathematical packages for verification.

The paper presents the case study of the industrial NPP I&C system, manufactured by RPC Radiy. The paper shows an application of metric-based approach and MSMC tool for dependability and safety analysis of RTS, and procedure of results verification.

1. INTRODUCTION

The dependability and safety assessment of the I&Cs which are used in safety domain, is an essential part of the development and certification process.

The need for high accurate assessment of both safety and dependability measures is strengthen by the I&Cs application area. For instance, the I&Cs used on NPP perform the following actions related to safety:

- provide to the operator an accurate and appropriate information and permit judicious action during both normal and abnormal operations;
- automatically control the main plant and many ancillary systems in different modes including emergency;

- protect the plant from the consequences of any mistakes, which the operator or the automatic control system may make;

- under abnormal conditions I&Cs provide rapid automatic action to protect both plant and the environment (IAEA, 1999).

One of the main standards in the safety domain IEC 61508-1 provides the requirements for functional safety measures – *PFDavg* (the average probability of dangerous failure on demand) and *PFH* (probability of failure per hour) based on the system safety integrity level (SIL) (IEC 61508, 2010).

One of the cost-effective solutions on dependability and safety assessment of the NPP I&Cs is model-based evaluation, as it is allows system evaluation without having to build and measure a systems. Such model-based evaluation can be performed through discrete-event simulation or analytic models. Both approaches have drawbacks, since simulation can estimate results only up to a certain level of accuracy (Buchholz, 2004) and analytical model tend to be more abstract because of additional assumptions, which are set to make models tractable. The analytical models can be split into two groups: state-space (Markov's chains, semi-Markov processes, Markov regenerative processes, SAN, etc.) and combinatorial (RBD, FTA, ect.) models. The state-space models are always preferred to non-space model, because they can easily incorporate realistic system behavior such as imperfect fault coverage, multiple failure modes, hot-swap components (Smith, 2008).

MC are well-know and widely applied in dependability and performability analysis of safety-critical systems, because of the ease and flexible representation of system components dependencies, synchronization and complex maintenance strategies, such as recover priority, limited recovery resource, etc. (IEC 61165, 2008). The basic property of MC is following: the knowledge of the probabilities of the system states at a given instant of time summarizes all the past and is enough to calculate how the system evolves in future can be very useful for *PFDavg* and *PFH* calculations (IEC 61508, 2010).

There are few roadblocks for even greater application of the MC: accounting the additional system components exponentially increases the model state space and complicates analysis; the non-numerically sophisticated user may find it difficult to decide between the variety of numerical solution methods and tools to determine the most suitable and accurate for their application (Kharchenko, 2014). The numerical methods are limited by model size (largeness) and such very essential MC features as stiffness (Trivedi, 1994) and sparsity (Reibman, 1988).

There are at least three important considerations for making decision between different solution techniques: efficiency and applicability of an algorithm, the structure of a matrix, size and storage needs (Barge, 2002). The largeness property forces to use additional storage place, while stiffness influence on the efficiency and applicability of a numerical solution algorithm and sparsity affects the structure of a matrix. Thus obtaining the high accurate and trusted modeling results becomes a nontrivial task.

The modeling experience (Kharchenko, 2013) goes in contrary with the recommendations from the one of the leading standards in the safety area IEC 61508-2010 (6th part), which asserts that "*efficient algorithms for solution of the MC were developed long time ago and implemented into software packages, so the modeler needs to focus only on the building of the model and not on the underlying mathematics*".

The previous research work (Kharchenko, 2014) shows that automated selection of the solution method based on the analysis of the main MC features can support not only the process of decision making between a wide set of approaches and methods, but also help in saving time, computational resources and can decrease the assessment risks.

In this paper, we present the metric-based approach for selection of the applicable solution technique, based on the analysis of MCs stiffness, largeness (decomposability, irreducibility), sparsity and fragmentedness. Using this selection procedure the modeler can also provide the verification of the earlier obtained results. The presented approach was implemented in the utility MSMC, which supports the MC construction, metric-based analysis, recommendations shaping and model solution. The model can be exported to the well-known off-the-shelf mathematical packages in verification purpose.

The case study using the proposed technique was performed over the industrial NPP I&C system, manufactured by RPC Radiy. This is a two-channel FPGA-based Reactor Trip System with three parallel chassis on voting logic "2-out-of-3" in each channel. The paper presents an application of the metric-based approach and MSMC tool for dependability and safety analysis of reactor trip system (RTS), and procedure of obtained results verification.

2. METRIC-BASED APPROACH FOR SELECTION OF MC SOLUTION TECHNIQUE

2.1 Test 1: Stiffness

Stiffness is well-known undesirable property of many practical MCs as it poses a problem of finding the transient solutions. Stiffness in models is caused by (Bobbio, 1986):

- in case of reparable systems the rates of failure and repair differ be several order of magnitude;

- fault-tolerant computer systems use redundancy, thus the rates of simultaneous failure of redundant components are typically significantly lower than failure rates of individual components;

- in models of reliability of modular software the modules' failure rates are significantly lower than the rates of passing the control from module to module.

The Cauchy problem problem *du/dx=F(x,u)* is said to be stiff on the interval *[x₀,X],* if there exists an *x* from this interval for which the following condition holds:

$$s(x) = \frac{\max\limits_{i=1,n} |\operatorname{Re}(\lambda_i)|}{\min\limits_{i=1,n} |\operatorname{Re}(\lambda_i)|} >> 1, \qquad (1)$$

where the *s(x)* – denotes the stiffness index and λ_i are the eigenvalues of a Jacobian matrix $(\operatorname{Re}\lambda_i < 0, i = 1,2,\ldots,n)$

(Arushanyan, 1990). The previous empirical work shows that quantitative value of s(x) have an impact on accuracy of different numerical methods – the higher *s(x)* value the more strict requirements imposed on the stability of chosen numerical method. Thus, we use *s(x)* as a main metric of stiffness. The *s(x)* values can be split into three groups: high $s(x) \geq 3*10^3$, moderate $10^2<s(x)<3*10^3$ and low $s(x)<10^2$ (Kharchenko, 2013).

The basic methods to overcome stiffness are described next.

Stiffness-avoidance approach. The basic idea of this approach is a model transformation by identifying and eliminating the stiffness from the model, which would bring two benefits: i) a reduction of the largeness of the initial MC, and ii) efficiency in solving a non-stiff model using standard numerical methods. The approach was named an aggregation/disaggregation technique for transient solution of stiff MCs. The technique, developed by K. S. Trivedi, A. Bobbio and A. Reibmann (Bobbio, 1986), can be applied to any MC with transition rates that can be grouped into two separate sets of values – the set of *slow* and the set of *fast* states. While the transformation of the initial stiff MC brings benefits in terms of efficiency, to the best of our knowledge, no systematic study has been undertaken of the impact of the transformation (from a stiff to a non-stiff MC on the accuracy of the solution.

Stiffness-tolerance approach. The main idea of this approach is using methods that are stable for solving stiff models. These methods can be split broadly into two classes: “classical” numerical methods for solution of stiff differential equations (DEs) and “modified” numerical methods used for finding a solution in special cases. Based on the analysis of earlier research works (Kharchenko, 2013, Butenko, 2014) and conducted empirical tests for each group of *s(x)* were selected the main MC solution technique and technique for results verification (Kharchenko, 2014). The Fig.1 shows normalized scale of *s(x)* with corresponding recommendations for method selection, where:

- *STA* – stiffness-tolerance approach;
- *SAA* – stiffness-avoidance approach;
- *m* – subscript, which denotes the main approach;
- *v* – subscript, which denoted the verification approach.

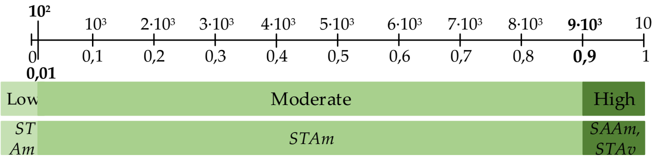


FIG. 1 NORMALIZED SCALE FOR STIFFNESS METRIC

The top values on scale (Fig. 1) are actual *s(x)* that are gained using (Eq. 1) and lower symbols shows the corresponding to them normalized values on interval [0; 1]. Normalization can be performed using (Eq. 2):

$$m_i = \frac{x_i - x_i^{\max}}{x_i^{\max} - x_i^{\min}}, \tag{2}$$

where m_i – normalized value, x_i – initial value, x_i^{max} - maximum on scale, for *s(x)* $x^{max} = 10^4$, x_i^{min} - minimum on scale, for *s(x)* $x^{min} = 0$.

2.2 Test 2: Largeness (Decomposability, Irreducibility)

MM of realistic systems are usually plagued by largeness of state space. In this case, the researched system is specified using some high-level formalisms, such as Petri nets and using this specification the underlying MC is generated. The basic solution methods for large MC are described next.

Largeness avoidance approach. The main idea of this approach is to avoid generation of the large MC from the beginning. Using largeness avoidance approach (LAA) approach the certain properties of model representation are exploited to reduce the size of the MC to obtain the measures of interest (Sanders, 2003). The state-level and model-level (Sanders, 2003) lumping techniques are well-known methods of LAA approach. A state-level lumping technique is a technique that exploits the certain properties on the MC level, while the model-level lumping denotes the lumping properties on the high-level formalism and directly construct lumped MC. Another LAA technique is an aggregation, which set a condition for partition of the state space, and replacing the formed sub-sets by a single state. The aggregation in contrast to lumping gives approximate results, with or without bounds, but may result the smaller MC then a lumping technique.

Largeness tolerance approach. The largeness tolerance approach (LTA) are designed to manipulate large MC using special algorithms and data structures to reduce and store transition probabilities matrix (Srinivasan, 1990). The numerous works (Srinivasan, 1990, Bryant, 1989) present the ideas of using binary and multi-valued decision diagrams (BDD and MDD), matrix diagrams (MD), Kronecker products, etc. to deal with state space size. The disk-based approach for steady-state and path-based approach for transient solutions are also considered in (Gail, 1989, Sanders, 1998). Analysis of MC irreducibility and decomposability properties can help to make a prior selection between described techniques (Barge, 2002).

The aggregation techniques are mainly based on the decomposability approach. In this case the *degree of coupling* can be taken as measure of matrix decomposability property. For example, considering a nearly completely decomposable (NCD) MC (Eq. 3), which has a matrix with non-zero elements in off-diagonal blocks are small compared with those in the diagonal blocks (Courtois, 1977):

$$A = \begin{pmatrix} A_{11} & A_{12} & \dots & A_{1n} \\ A_{21} & A_{22} & \dots & A_{2n} \\ \dots & \dots & \dots & \dots \\ A_{n1} & A_{n2} & \dots & A_{nn} \end{pmatrix}, \tag{3}$$

where $A_{11}, A_{12}, \dots, A_{nn}$ are square diagonal subblocks. The stationary distribution of π can be partitioned such as $\pi = (\pi_1, \pi_2, \dots, \pi_n)$. Assuming that A is of form (Eq. 4), where E contains all of-diagonal blocks. The quantity (Eq. 5) is referred to as degree of decomposability (Courtois, 1977). If $E = 0$ then MC is said to be completely decomposable (CD).

$$A = diag(A_{11}, A_{22}, \dots, A_{nn}) + E, \tag{4}$$

$$\|E\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |e_{ij}| \tag{5}$$

An irreducible MC is presented by a direct graph that is a single strongly connected component. The algorithm for determining strongly connected components is a known graph algorithm (Cormen, 2001). Detection of such components (irreducibility property) can help in determining sub-sets for approximate aggregation technique, but it naturally applied after selecting the avoidance approach.

The value (Eq. 5) can help in deciding between avoidance and tolerance approaches, thus we refer to (Eq. 5) as a main largeness metric, further decomposability metric. The *E* values can be split into three groups (Kharchenko, 2014):

- completely decomposable (CD): $E < 0.3$;
- nearly completely decomposable (NCD): $0.3 \leq E < 0.6$;
- non-decomposable (ND): $E \geq 0.6$.

As in the previous test, Fig.2 presents normalized scale for *E* with recommendations for approach selection, where:

- *LTA* – largeness-tolerance approach;
- *LAA* – largeness-avoidance approach;
- *m* and *v* denote the main and verification approach, respectively;
- $x^{max} = 0.9$ and $x^{min} = 0$.

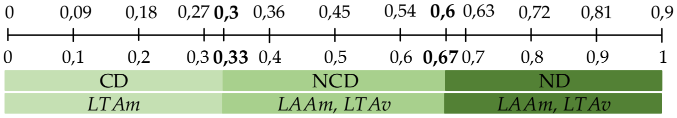


FIG. 2 NORMALIZED SCALE FOR DECOMPOSABILITY METRIC

2.3 Test 3: Sparsity

Modeling the components interaction enlarges the MC state space significantly, thus the sparse structures are required. Transient solution methods that do not preserve sparsity are unacceptable for most large problems (Reibman, 1988). The direct methods for finding the steady-state solutions in case of sparse matrices may depend on the common sparse patterns, such as band/block diagonal forms, band/block tridiagonal, cyclic banded forms, etc. (Press, 2007). Paper (Barge, 2002) presents the formula for evaluation of the heuristic measure of sparsity – matrix score (Eq. 5). It gives a measure of how the matrix elements are dispersed from the main diagonal.

Let q_i be the number of matrix elements that are a distance *i* from the diagonal. The histogram is weighted and then scaled by n^2 where *n* is matrix order. The matrix score *ms* can be evaluated using (Eq. 6):

$$ms = \left(\sum_{i=1}^{n-1} i \cdot q_i\right) / n^2 \qquad (6)$$

In (Barge, 2002) authors studied the influence of *ms* value on accuracy of the tolerance techniques for MC solution, and recommended to give additional attention on *fill-in* amount for matrices with $n \geq 500$ and $ms > 0.8$. The *fill-in* is a property when initially zero matrix elements become nonzero during solution process and for which storage must be reserved.

The value *ms* can be classified into three groups (Kharchenko, 2014):

- high sparsity: $ms < 0.3$;
- moderate sparsity: $0.3 \leq ms < 0.72$;
- low sparsity: $ms \geq 0.72$.

Fig. 3 presents normalized scale of *ms* with recommendations for approach selection, where $x^{max} = 0.9$ and $x^{min} = 0$.

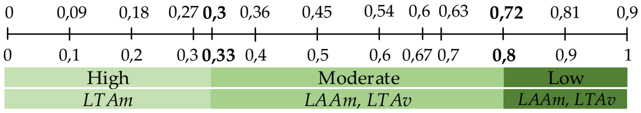


FIG. 3 NORMALIZED SCALE FOR SPARSITY METRIC

2.4 Test 4: Fragmentedness

The MC are widely applied to analyze the dependability of physical components of safety-critical systems. Assessing the dependability of software components depends on how well the component has been tested, is it available and whether it is a reused or new component (Kharchenko, 2011). The verification and validation (V&V) phases are strongly managed by requirements and recommendations of international standards (see standards IEC 60800-2006, IEC 61508 - 2010). There are lots of required procedures to test the software component, such as documentation analysis, problem review, static code analysis, etc (Butenko, 2014). Nevertheless, the residual software bugs can appear during system operation. In this case, to predict the general system dependability we need to observe not only hardware and software components separately, but also analyze their interconnection and total influence on system dependability.

In the previous research works (Kharchenko, 2014, Butenko, 2014) we applied the *multi-fragmentation principle* to present such complex interconnection between system hardware and software. The main idea of this approach is to capture and represent using MC the plausible phenomenon – variation of software failure – that is well accepted on practice (Kharchenko, 2013). Using this principle the model can be divided into N_{fr} fragments that are with the same structure but may differ in one or more parameters.

The number of fragments N_{fr} in MC depends on the number of expected undetected software faults n_i in *i*-different software versions (Eq. 7):

$$N_{fr} = \prod_{i=1}^{m} (n_i + 1) \qquad (7)$$

The structure of fragment and number of such fragments can help to determine the complexity of resulting multi-fragmental model (MFM) and thus help in making decision between avoidance and tolerance approaches. In this paper, we use N_{fr} as a metric of fragmentedness.

Based on N_{fr} value the MC can be classified as follows:

- low-fragmented: $N_{fr} < 6$;
- moderately fragmented: $6 \leq N_{fr} < 15$;
- highly fragmented: $N_{fr} \geq 15$.

The normalized scale ($x^{max} = 30$ and $x^{min} = 0$) with recommendations for approach selection is presented on Fig. 4.

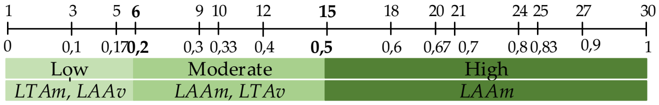


FIG. 4 SCALE FOR FRAGMENTEDNESS METRIC

2.5 Metric-Based Diagram

The recommendations for selecting the solution approach presented on fig. 1 – 4 were received separately for each characteristic. However, it is important to consider each MC feature, while making decision on the most efficient technique. In this section we present the metric-based diagram (Fig. 5), that incorporates all MC characteristics and supports the approach selection based on some specific combination of *s(x)*, *E*, *ms* and N_{fr} values. The diagram passed verification on the wide range of MCs.

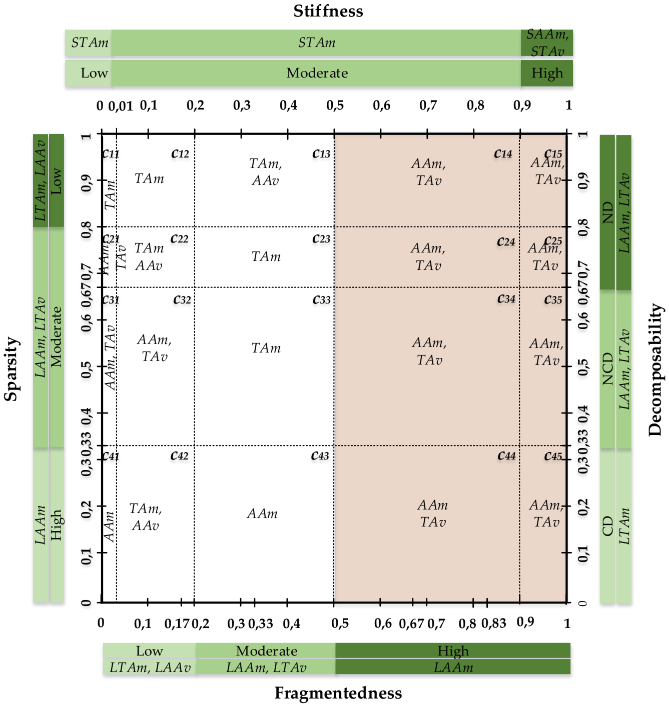


FIG. 5 METRIC-BASED DIAGRAM

The diagram is applied in three main stages.

1. Calculation of stiffness, decomposability, sparsity and fragmentedness metrics using (Eq. 1), (Eq. 5) – (Eq. 7).
2. Normalization of the received values using (Eq. 2).
3. Marking of the normalized metrics values on the appropriate scale and creating the intersection by drawing perpendicular to the opposite side. As a result, we receive the rectangle, placed in one of the internal zones $c_{ij}, i\in(\overline{1,4}), j\in(\overline{1,5})$. Each zone provides the recommendation for selection of avoidance or tolerance approach, *AA* or *TA* respectively. If rectangle is placed in two or more zones, selecting the recommendations from zone that contains the larger area of the rectangle. The colored inner zones define the SAA or STA use.

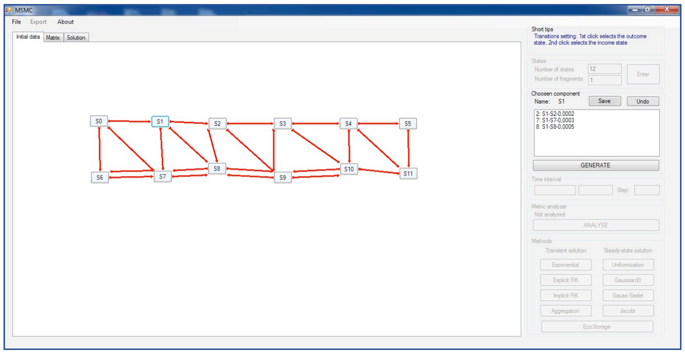
3 MSMC TOOL

The "MSMC – Method selector for Markov chains" was developed to help user, unsophisticated in Markov modeling to select the most effective solution approach automatically, which will give an accurate result. The metric-based approach described in section 2 was implemented in MSMC. This is an alpha version, which runs on Windows platform (XP and later).

MSMC supports:

1. Two types of MC construction: graphical and analytical.
2. Stiffness, sparsity, decomposability and fragmentedness testing.
3. Selection of the solution approach using the tests results.
4. State probabilities calculation using selected approach.
5. Conversion of transition probabilities matrix and DES into the MATLAB, Mathematica and Maple syntaxes in verification purpose.

The major components of a GUI are *model editor*, *transitions editor*, *generated matrix* tab, *solution* tab and *methods* panel. The model editor (Fig. 6) allows a graphical construction of the MC using MSMC primitives – places (states) and arcs (transitions). If MC already exists user can upload the transition probabilities matrix as an Excel spreadsheet (.xls or .xslx) or text (.txt) document. With the transitions editor user can change already assigned transition values.



Short tips
Transitions setting: 1st click selects the outcome state, 2nd click selects the income state

States
Number of states 12
Number of fragments 1

Choosen component
Name: S1

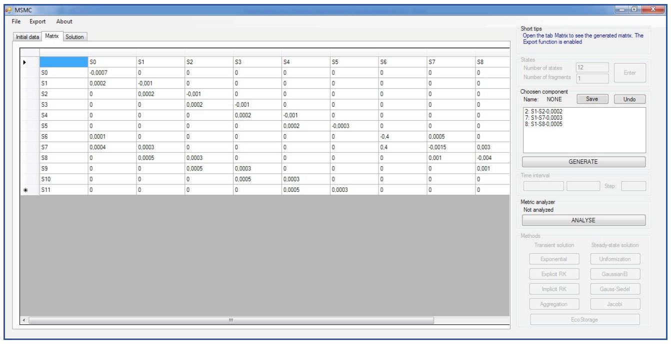
2: S1-S2-0,0002
7: S1-S7-0,0003
8: S1-S8-0,0005

Time interval
Step:

Metric analyzer
Not analyzed

Methods
Transient solution
Steady-state solution

FIG. 6 MC GRAPHICAL CONSTRUCTION



	S0	S1	S2	S3	S4	S5	S6	S7	S8
S0	-0,0007	0	0	0	0	0	0	0	0
S1	0,0002	-0,001	0	0	0	0	0	0	0
S2	0	0,0002	-0,001	0	0	0	0	0	0
S3	0	0	0,0002	-0,001	0	0	0	0	0
S4	0	0	0	0,0002	-0,001	0	0	0	0
S5	0	0	0	0	0,0002	-0,0003	0	0	0
S6	0,0001	0	0	0	0	0	-0,4	0,0005	0
S7	0,0004	0,0003	0	0	0	0	0,4	-0,0015	0,003
S8	0	0,0005	0,0003	0	0	0	0	0,001	-0,004
S9	0	0	0,0005	0,0003	0	0	0	0	0,001
S10	0	0	0	0,0005	0,0003	0	0	0	0
S11	0	0	0	0	0,0005	0,0003	0	0	0

Short tips
Open the tab Matrix to see the generated matrix. The Export function is enabled

States
Number of states 12
Number of fragments 1

Choosen component
Name: NONE

2: S1-S2-0,0002
7: S1-S7-0,0003
8: S1-S8-0,0005

Time interval
Step:

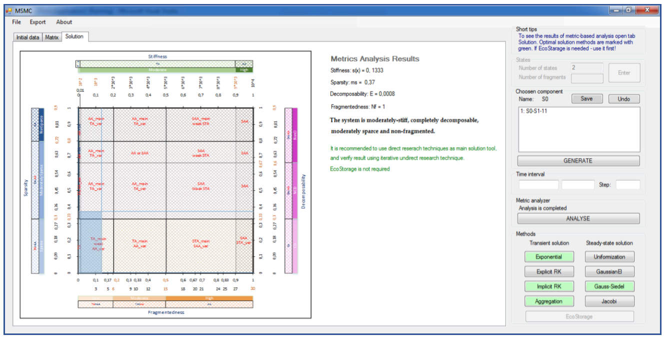
Metric analyzer
Not analyzed

Methods
Transient solution
Steady-state solution

FIG. 7 GENERATED TRANSITION PROBABILITIES MATRIX

After the MC construction, MSMC generates an underlying transition probabilities matrix and see the result in *generated matrix* tab (Fig. 7). With the *Export* option user can convert the received matrix into MATLAB, Mathematica and Maple form and save it as text file.

The stiffness, decomposability, sparsity and fragmentedness test results, metric-based diagram and recommendations for approach selection are displayed in *solution* tab (Fig. 8).



Metrics Analysis Results

Stiffness: s(x) = 0, 1333

Sparsity: ms = 0,37

Decomposability: E = 0,0008

Fragmentedness: Nf = 1

The system is moderately-stiff, completely decomposable, moderately sparce and non-fragmented.

It is recommended to use direct reserach techniques as main solution tool, and verify result using iterative undirect research technique.

EcoStorage is not required

Short tips
To see the results of metric-based analysis open tab Solution. Optimal solution methods are marked with green. If EcoStarage is needed - use it first!

States
Number of states: 2
Number of fragments:

Choosen component
Name: S0
1: S0-S1-11

Time interval
Step:

Metric analyzer
Analysis is completed

Methods
Transient solution: Exponential, Explicit RK, Implicit RK, Aggregation
Steady-state solution: Uniformization, GaussianEl, Gauss-Siedel, Jacobi

FIG. 8 METRIC-BASED APPROACH APPLICATION

If the matrix appears to be moderately or highly sparse MSMC will store the non-zero elements in flat format using the algorithm described in (Press, 2007). Thus, all subsequences of calculations are performed on the compressed matrix form. We have implemented the well-known numerical methods (Press, 2007) (implicit RK, explicit RK, exponential, etc.) and aggregation techniques (Bobbio, 1986) to find the MC transient solution. Using the methods panel user can select the recommended method (-s), which are highlighted with color. The solution results are automatically saved into the text file.

4 CASE STUDY

Here we illustrate the application of metric-based approach and MSMC for dependability and safety assessment of typical NPP I&Cs. This is a Reactor Trip System constructed on the FPGA-based digital platform RadICS, produced by RPC Radiy.

Generally, the platform can contain up to 7 modules: analogue and digital input modules (AIM, DIM); analogue and digital output modules (AOM, DOM); logic module (LM); optical communication module (OCM); and analogue input for neutron flux measurement module (AIFM). The modules can be placed in 16 different positions on the platform (two reserved positions for LM), using LVDS and fiber optical lines for internal/external communications. Such flexible redundancy management helps to ensure the high availability of the system.

In section 4.1 we briefly describe the studied RTS and it reliability-block diagram, as the more detailed description is given in (Butenko, 2014). Section 4.2 presents the Markov chain for RTS and application of the metric-based approach. In section 4.3 shows the results of dependability and safety parameters assessment using recommended methods in MSMC tool. Obtained results were verified with build-in functions in few off-the-shelf mathematical packages.

4.1 Description of the System under Study

The observed RTS is a two-channel, three-chassis architecture, with voting logic “2-out-of-3” for chassis in each channel and “1-out-of-2” between channels. Both channel independently receive information from sensors and other NPP systems and capable to form the reactor trip signal. Each chassis, observed in this paper, consists of five modules: LM, DIM, DOM, AIM and AOM.

The RTS uses diverse software (Kharchenko, 2013), i.e. non-identical but functionally equivalent software copies are deployed on each channel.

The RTS reliability-block diagram is presented on Fig. 9. Reliability index *Ppfi.j* determines hardware reliability of the chassis *Ti.j* (defined by physical faults), where *i* indicates main (*T1.j*) or diverse (*T2.j*) channels, and *j* indicates the number of the chassis. Reliability index *Pdfi* determines software reliability of the main or diverse channels (defined by software faults), where *i* indicates channel. Reliability index *Pmi*, determines reliability of the majority element *mi*, where $i \in (1,3)$.

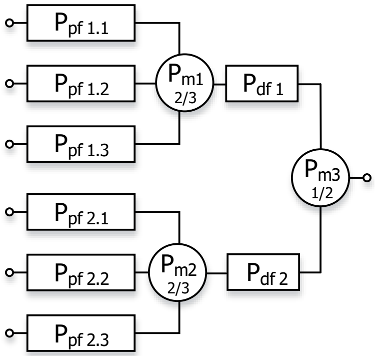


FIG. 9 RELIABILITY-BLOCK DIAGRAMM OF RTS

4.2 Markov Chain of the RTS

Let us further denote by λ_d and λ_p the design and physical failure rates, respectively, and by μ_d and μ_p the repair rates after design and physical failures. The MC for RTS system is shown on Fig. 11. The detailed model construction and description is shown in (Butenko, 2014).

We use the following assumptions to create the MC for observed RTS:

1. Each element of the research system in random moment of time can be only in two states – working and failure.

2. The systems control and majority elements provide unstoppable correct functioning.

3. The system maintenance is performed by one group of engineers, thus failed chassis are repaired sequentially. It should be noted, that recovering strategy use case of two working channels in a priority.

4. All detected defects are eliminated instantaneously and no new defects are introduced. The mean time between failures and mean time to repair are exponentially distributed (Butenko, 2014).

5. Software testing datasets are updated after each test. The testing is performed on the complete body of input data.

6. The observed RTS is FPGA-based, thus investigated software faults are such kinds of faults, which are typical for VHDL coding process that were not covered by V&V procedure.

The architecture-level MC shows the rare kind of design faults that can cause a general system failure, thus we expect that not more than two undetected design faults on each software version (Ehrlich, 1990, Butenko, 2014).

7. The failure rate of the design faults $\lambda_{d(i)}$ is proportional to their residual amount n_i in i – different software versions

(Butenko, 2014). This assumption uses an incremental change of the software failure rate after detected design fault elimination ($\lambda_{d(i)}$ vary on a constant $\Delta\lambda_{d(i)}$). Such failure rates can be presented using multi-fragmentation approach (Kharchenko, 2011).

8. The design failures on diverse software versions are independent events, but equal in severity. Thus, we assume that failure and repair rates for the failures caused by design faults are equal (Eq. 8) – (Eq. 9).

$$\lambda_{d1} = \lambda_{d2} \Rightarrow \lambda_d = \lambda_{d1} + \lambda_{d2}, \qquad (8)$$

$$\mu_{d1} = \mu_{d2}; \mu_d = \lambda_d / (\sum_{i=1}^{2} \frac{\lambda_{d(i)}}{\mu_{d(i)}}) \qquad (9)$$

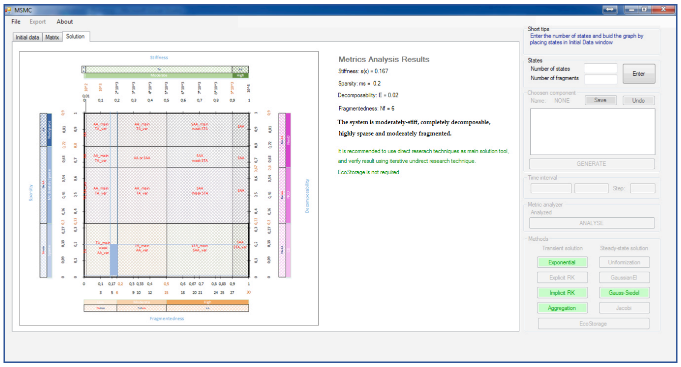
The neglecting of this assumption will increase the resulting MC state-space, but still we can apply the MFM principle for it construction. Thereby, the assumption was used in a purpose of reducing the model size.

To check the developed model sensitivity we use four sets of parameters values, which are presented in Table 1.

TABLE 1. MC PARAMETER VALUES

	λ_d (1/h)	$\Delta\lambda_d$ (1/h)	λ_p (1/h)	μ_d (1/h)	μ_p (1/h)	t (h)
1	10^{-5}	$5 \cdot 10^{-6}$	10^{-4}	0.01	1	[0; 30 000]
2	$2.5 \cdot 10^{-5}$	$1.25 \cdot 10^{-5}$				
3	$5 \cdot 10^{-5}$	$2.5 \cdot 10^{-5}$				
4	$7.5 \cdot 10^{-5}$	$3.75 \cdot 10^{-5}$				

The results of testing the stiffness (Eq. 1), decomposability (Eq. 5), sparsity (Eq. 6) and fragmentedness (Eq. 7) characteristics are as follows.



Enter the number of states and buid the graph by placing states in Initial Data window

Metrics Analysis Results

Stiffness: s(x) = 0.167

Sparsity: ms = 0.2

Decomposability: E = 0.02

Fragmentedness: Nf = 6

The system is moderately-stiff, completely decomposable, highly sparse and moderately fragmented.

It is recommended to use direct reserach techniques as main solution tool, and verify result using iterative undirect research technique.

EcoStorage is not required

FIG. 10 METRIC-BASED DIAGMRAM FOR RTS

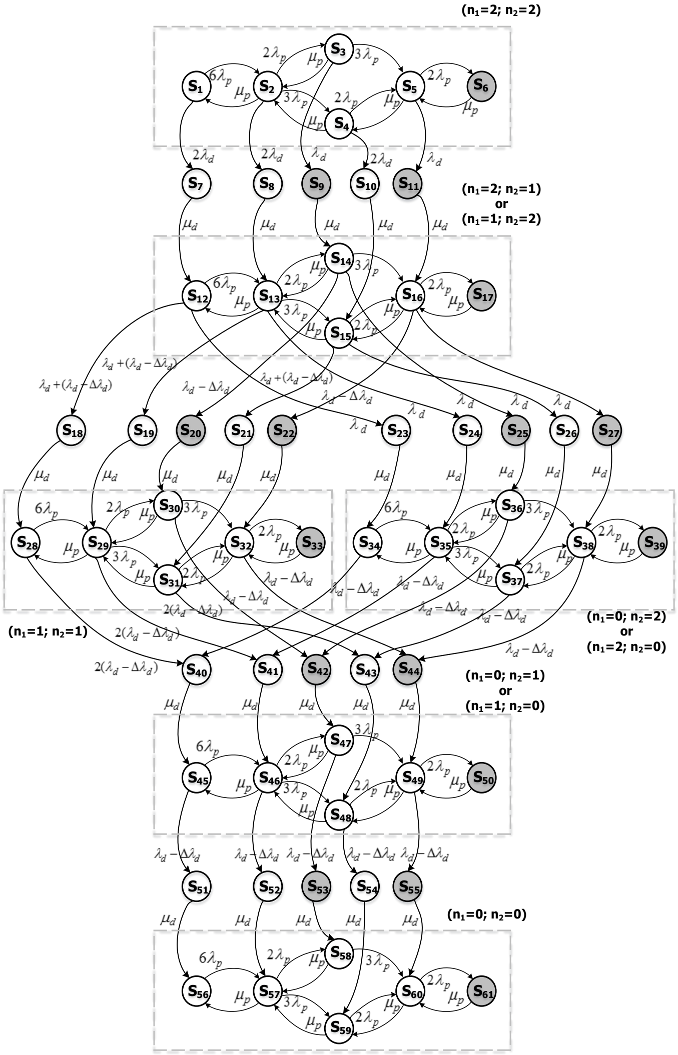


FIG. 11 MULTI-FRAGMENTED MARKOV MODEL OF RTS

1. Stiffness: moderately stiff with $s(x) = 0.167$.

2. Decomposability: completely decomposable (CD) with $E = 0.02$.

3. Sparsity: highly sparse with $ms = 0.2$.

4. Fragmentedness: moderately fragmented with $N_{fr}=6$.

With the metric-based diagram (Fig. 10), we receive the recommendation to use tolerance approach as the main solution technique and verify obtained result with avoidance approach. As the model appears to be moderately stiff, we need to use stiffness-stable numerical methods (STA).

4.3 Solution and Results Verification

We use the recommended approaches to assess the RTS unavailability function, which also defines the *PFH* measure (IEC 61508, 2010).

The unavailability function *U(t)* is defined as a sum of failed states probabilities, with initial condition *U(0) = 0* (Eq. 10):

$$U(t) = 1 - A(t) = \sum_{i=1}^{n} P_i(t), i \in N, \qquad (10)$$

where, *A(t)* is RTS availability function.

Due to recommendations we use the STA as a main techniques, thus the *U(t)* for four sets of RTS parameter (Table 1) was calculated using build-in function of implicit RK in Mathematica. The results are shown on Fig. 12.

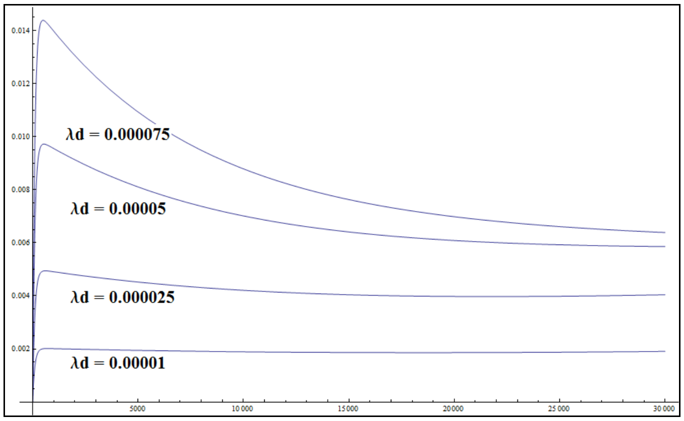


FIG. 12 U(t) CALCULATED USING STA

The *U(t)* result were verified using MSMC inner function, that implements the implicit RK algorithm and SAA, particularly aggregation/ disaggregation technique (Bobbio, 1986). The result of *U(t)* verification for $\lambda_d = 5 \cdot 10^{-5}$ is presented on Fig. 13.

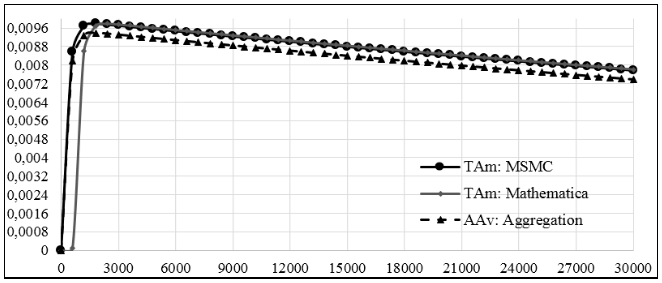


FIG. 13 U(t) RESULTS VERIFICATION

5. CONCLUSION

This paper presents the metric-based approach for selection of the applicable solution approach, based on the analysis of such MCs characteristics as stiffness, largeness (decomposability, irreducibility), sparsity and fragmentedness. The metric-based approach was implemented into MSMC tool that support the graphical construction of MC, with further generation of underlying matrix and testing its properties. Based on the test results MSMC tool automatically presents the recommendations for approach selection and provides the transient solution using inner numerical methods.

The paper describes a case study for assessment the unavailability parameter of NPP I&C system, in particular, RTS. We present the RBD and MC system models and use the multi-fragmentation principle to describe the complex hardware-software interconnection on the architecture level. The metric-based approach was applied to make an informed selection of the solution approach. We have tested the RTS unavailability function for different parameter sets (Table 1), using the main recommended approach – STA. The results were also verified by SAA. Based on the obtained U(t) values we can conclude that under all assumption presented in Section 4, the studied architecture of RTS constructed on the FPGA-based digital platform, provides the needed dependability and safety level.

In our future work we intend to calculate the risk function for presented RTS architecture, by eliminating the assumption that design failures are equal in severity and to finalize development of tool supporting MC-based safety assessment using suggested metric approach and procedure of technique and tool selection, that minimize the risks of inaccurate calculations.

NOMENCLATURE

CD	Completely decomposable
FPGA	Field programmable gate array
FTA	Fault trees analysis
GUI	Graphical user interface
I&Cs	Instrumentation and Control System
LAA	Largeness avoidance approach
LTA	Largeness tolerance approach
MC	Markov chain
MFM	Multi-fragmental model
MSMC	Method selector for Markov chains
NCD	Nearly completely decomposable
ND	Non-decomposable
NPP	Nuclear Power Plant
PFDavg	Average probability of dangerous failure on demand
PFH	Probability of failure per hour
RBD	Reliability block diagram
RK	Runge-Kutta
RTS	Reactor Trip System
SAA	Stiffness avoidance approach
SAN	Stochastic activity nets
SIL	Safety integrity level
STA	Stiffness tolerance approach

REFERENCES

IAEA, 1999, “Modern Instrumentation and control for NPP: A guidebook”, No. 387.

Buchholz, P., et al., 2004, “Approximate Computation of Transient Results for Large MC”, In proc. of IEEE QUEST’04, pp. 126-135.

Smith, W. E., et al., 2008, “Availability Analysis of Blade Server Systems”, IBM Systems Journal, Vol. 47(4), pp. 1- 20.

IEC 61165, 2008, “Application of Markov techniques”.

IEC 61508, 2010, “Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems”.

Kharchenko, V., et. al., 2014, “Markov’s Model and Tool-Based Assessment of Safety-Critical I&C Systems: Gaps of the IEC 61508”, In Proc. 12th Int. Conf. PSAM, p. 16.

Trivedi, K.S., et al., 1994, “Stiffness-Tolerant Methods for Transient Analysis of Stiff Markov Chains”, Microelectronic Reliability, Vol.34(11), pp.1825-1841.

Reibman, A., et al., 1988, “Numerical Transient Analysis of Markov models”, Comput. Opns. Res., Vol.15(1), pp. 19-36.

Barge, W. S., et al., 2002, “Autonomous Solution Methods for Large Markov Chains”, Pennsylvania State University CiteSeerX Archives, p. 17.

Kharchenko, V., et. al., 2013, “Availability Assessment of Computer Systems Described by Stiff Markov Chains: Case Study”, Springer, CCIS(412), pp. 112 – 135.

Bobbio, A., et al., 1986, “A Aggregation Technique for Transient Analysis of Stiff Markov Chains”, IEEE Transactions on Computers, C-35, pp. 803-814.

Arushanyan, O., et. al., 1990, “Numerical Solution of Ordinary Differential Equations using FORTRAN”, Moscow State University, Moscow, p. 336.

Butenko, V., 2014, “Modeling of a Reactor Trip System Using Markov Chains: Case Study”, Proc. of 2014 22nd ICONE, Vol. 5.

Sanders, W. H., et al., 2003, “Optimal State-space Lumping in Markov Chains”, Inf. Process. Lett., Vol. 87(6), pp. 309-315.

Srinivasan, A., et al., 1990, “Algorithms for Discrete Functions Manipulation”, In Proc. Int’l Conf. on CAD (ICCAD’90), pp. 92-95.

Bryant, R. E, 1986, “Graph-based Algorithms for Boolean Function Manipulation”, IEEE Trans. Comp., Vol. 35(8), pp. 677–691.

Gail, H. R., et al., 1989, “Calculating Availability and Performability Measures of Repairable Computer Systems”, Journal of the ACM, Vol. 36, pp. 171–193.

Sanders, W. H., et al., 1998, “An Efficient Disk-based Tool for Solving Large Markov Models”, Performance Evaluation, Vol. 33, pp. 67–84.

Courtois, P. J., 1977, “Decomposability: Queueing and Computer Applications”, Academic Press, New York, p. 201.

Cormen, T., 2001 “Introduction to Algorithms”, MIT Press, Computers, p. 180.

Press, W.H., et al., 2007, “Numerical Recipes. The Art of Scientific Computing, 3rd Edition”, Cambridge University Press, p. 1260.

Kharchenko, V., et al., 2011, “Multi-fragmental Availability Models of Critical Infrastructures with Variable Parameters of System Dependability”, Int. Journal Information & Security, Vol 28. pp. 248 – 265.

Ehrlich, W., et al., 1990, "Applying Reliability Measurement: A Case Study". IEEE Software, March 1990, pp. 56-64