Application of Markov Modeling for Safety Assessment of Self-Diagnostic Programmable Instrumentations and Control Systems

Valentyna Butenko, Oleg Odarushchenko, Vyacheslav Kharchenko, Viktoriya Moskalets,
Elena Odarushchenko, Oleksii Strjuk

Abstract**—**Markov modeling is a well-known analytical state space modeling technique which is widely applied for quantitative analysis of safety-critical systems. There are few roadblocks for greater application of Markov modeling: accounting of additional system components increases the model state space and complicates analysis; the non-numerically sophisticated user may find it difficult to select method and tool to provide an accurate analysis of constructed Markov model. Thus, achieving highly trusted result for safety-critical systems is a nontrivial task. In this paper we present the case-study on application of Markov modeling with deep testing the model features, for safety analysis of industrial self-diagnostic, programmable FPGA-based Instrumentation and Control system which operates on Nuclear Power Plant.

Keywords**—**Instrumentation and Control system, Reactor Trip System, Markov model, metric-based approach

I. INTRODUCTION

An accurate safety assessment is a key task during development and certification of safetycritical systems as it allows to demonstrate that relevant requirements have been met. Detailed safety analysis is extremely important in case of design of Instrumentation and Control systems (I&Cs) that function on Nuclear Power Plants (NPP) because of potential risks for environment and people. To prevent any accident that may occur during continuous work of I&Cs various software and hardware diversity architectures were developed as well as deep self-diagnostic methods [1].

Normal operating and emergency protection systems are typical examples of NPP I&Cs.

The range of approaches were standardized to determine how to achieve the high accuracy goal. Among such widely applied techniques as FMEA, FMECA, FMEDA etc. stands Markov modeling. This is a well-known analytical state space modeling technique that was applied through decades to assess mainly dependability of safety-critical systems, meanwhile it is stated as one that can be applied for safety assessment as well [2]. With state space approaches the modeler can analyze failure/repair dependencies, shared repair facilities [3] results of errors in self-diagnostic tools and provide the detailed presentation of system behavior for communication with engineering team [4].

Modeling components interaction and interdependencies expands the model significantly, thus making the precise computation of system transient measures almost infeasible. Whilst numerical methods and imitation modeling can be applied to handling this problem, they are also limited by model size, also known as *largeness,* and such difficulties as *stiffness* [5] and *sparsity* [6]. Stiffness is an undesirable property of many practical Markov models (MM) and usually it is caused by: i) in case of repairable systems the rates of failure and repair differ by several orders of magnitude; ii) fault-tolerant computer systems (CS) use redundancy. The rates of simultaneous failure of redundant components are typically significantly lower than the rates

V. Kharchenko, Centre for Safety Infrastructure Oriented Research and Analysis, Kharkiv, Ukraine (e-mail: v.kharchenko@csn.khai.edu)
O. Odarushchenko, Research and Production Company "Radiy", Kirovograd, Ukraine (e-mail: odarushchenko@gmail.com)
V. Butenko, National Aerospace University "KhAI", Kharkiv, Ukraine (v.butenko@csn.khai.edu)
V. Moskalets, Research and Production Company "Radiy", Kirovograd, Ukraine (e-mail: moskalets.viktoriya@mail.ru)
E. Odarushchenko, Poltava National Technical University, Poltava, Ukraine (e-mail: skifs2007@mail.ru)
O. Strjuk, Research and Production Company "Radiy", Kirovograd, Ukraine (e-mail:strjuk.radiks@gmail.com)

of the individual components; iii) in models of reliability of modular software the modules' failure rates are significantly lower than the rates of passing the control from a module to a module [7]. Sparsity [8] corresponds to systems, which are loosely coupled. In the subfield of numerical analysis, a sparse matrix is a matrix populated primarily with zero's [9]. If the MM is large it becomes wasteful to reserve storage for zero elements, thus solution methods that do not preserve sparsity, is unacceptable for most large problems [6].

Variety of approaches were developed to deal with MM largeness, stiffness and sparsity. They can be split into two large groups – "avoidance" and "tolerance" techniques, each contains range of advantages and limitations, thus making methods acceptable within specific conditions. In [10, 11] authors have presented the detailed description of the most common techniques, highlighting both advantages and disadvantages of their use.

Manual computations are unacceptable for large state space models. The range of specialized, off-the-shelf tools and utilities were developed to support Markov modeling process making the assessment more convenient. Still such variety or tools and techniques (T&T) can pose a difficulty when it comes to choose the most appropriate set for a specific case as every T&T is limited in its properties and applicability. The careful selection is important in case of stiffness, largeness and sparsity presence in MM, as it requires the modeler to focus on math details to avoid the use of inefficient T&T.

One of the leading standards in the safety area IEC 61508-2010 provides no special requirements for T&T, which are used to evaluate the system safety indicators, excepting the strong recommendation, that practitioner must have an understanding of the techniques used by software package to ensure its use is suitable for the specific application [2]. In contrast to the T&T, many requirements were developed for I&Cs verification and validation (V&V) tools and they are compatible by strength to the requirements of produced software and systems (see standard IEC 60880-2006 [12]). Additionally, this standard (IEC 61508-2010) asserts that methods for solving Markov models have been developed long ago and trying to improve these methods does not seem sensible. The previous works show [13, 14] that solving a large and/or stiff Markov model requires a careful selection of the solution method/tool. Otherwise, the results can differ in several orders of magnitude [14], thus, use of inappropriate method/tool for the solution of a non-trivial MM may lead to significant errors. To ensure accuracy of selected T&T for a specific case, we have applied the metric-based approach [15] during safety analysis of self-diagnostic, programmable NPP I&Cs.

In this paper we present the case-study of typical safety assessment for self-diagnostic and programmable NPP I&Cs produced by RPC "Radiy". The section 2 shows main information on I&Cs elements, structure diagram and reliability-block diagram (RBD) of analyzed architecture. Section 3 presents the developed MM. Application of metric-based approach during safety analysis and achieved results are shown in Section 4. In section 5 we present the conclusions and problems left for future research.

II. I&C Architecture

This section presents the typical architecture of self-diagnostic and programmable NPP I&Cs produced by RPC "Radiy".

This is a Reactor Trip System (RTS) with two-channel, three-track architecture, on voting logic "2-out-of-3" for tracks in each channel and "1-out-of-2" between channels. The FPGAbased track is a basic component of observed RTS. Generally, each track can contain up to 7 module types: analogue and digital input modules (AIM, DIM); analogue and digital output modules (AOM, DOM); logic module (LM); optical communication module (OCM); and analog input for neutron flux measurement module (AIFM). The modules can be placed in 16 different positions on the track (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. Each channel independently receives information from sensors and other NPP systems. The channels, each being capable of forming a reactor trip signal, are independent.

The LM from each track is connected to self-diagnostic equipment, which constantly perform tests over the received data and informs on found failures and/or deviations. The diagnostic tests can only trace the found failures and have no influence on channels output data nor on general result.

In this paper, we consider the tracks consisting of five modules: LM, DIM, DOM, AIM and AOM. The Fig. 1 presents the structure diagram of a typical track. It is assumed that the corresponding components of all the tracks in the channels are identical, i.e. DIM on the 1st track is identical to the same module on other tracks in the channels, etc. The failure of the LM leads to the failure of the whole track, and failures of the DIM, DOM, AIM, AOM result in track malfunction. Therefore, it was assumed that failure of any module implies the general failed state of the track. The RDB for RTS is presented on Fig. 2. All tracks in the channels have identical hardware structures, but the software run on the system channels is diverse [16], i.e. non-identical but functionally equivalent software copies are deployed on the system channels.

Reliability index $P_{pf\,i.j}$ determines the hardware reliability of the track $T_{i.j}$ (defined by physical faults), where i indicated main ($T_{1.j}$) or diverse ($T_{2.j}$) channel, and j indicated the track number. Reliability index P_{dfi} determines software reliability of the main or diverse channels (defined by software faults), where i indicates the channel. Reliability index P_{mi}, determines reliability of majority element m_i, where $i \in (\overline{1,3})$ and D_l is the reliability of control and diagnostic tools, where $l \in \{1, 2\}$.

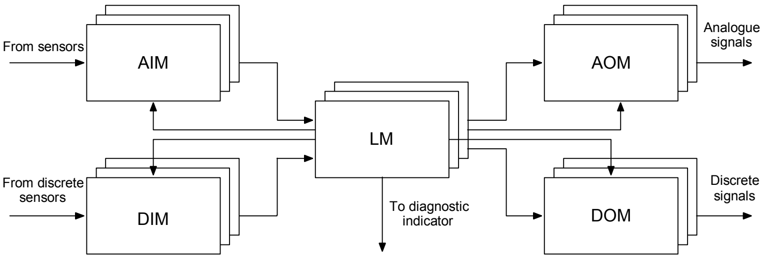

Fig.1 The structure diagram of the typical track

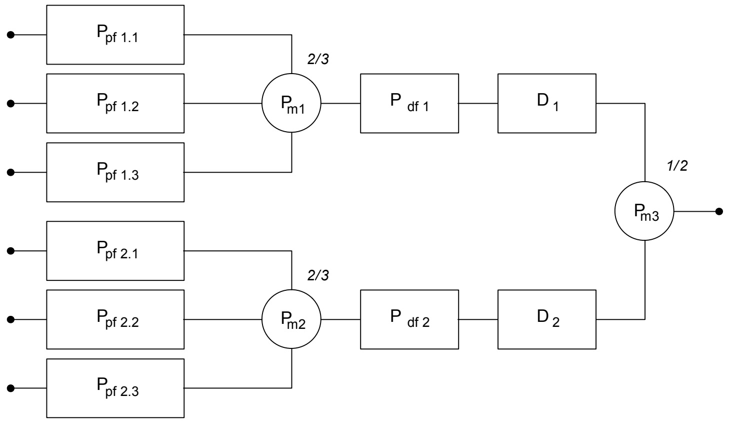

Fig.2 The reliability-block diagram of two-channel tree-track Reactor Trip System

III. Markov Model

A. Assumptions

The following assumptions were used to create the MM for observed RTS:

a. each element in the random moment of time can be only in two states – working and failure;

b. the system majority elements provide unstoppable correct functioning;

c. the self-diagnostic and control equipment is identical for both channels and tracks the physical faults occurrence;

d. the maintenance is performed by one group of engineers and failed chassis are repaired sequentially;

e. 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;

f. software testing datasets are updated after each test;

g. the design faults 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.

h. 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 MM shows the rare kind of design faults that can cause a general system failure, thus we expect that not more than one undetected design fault on each software version [17, 18];

i. we assume that the rates of failure and repair of software will vary over time, e.g. as a result of executing the software in partitions as discussed in [13]. We implement this assumption based on the research work [17] that shows a plausible phenomenon – variation of software failure rates - which is well accepted in practice.

B. Model Parameters

The MM use the following set of parameters:

a. $\lambda_{d(i)}$ – design faults failure rate, which is proportional to their residual amount n_i in i – different software versions. The change of residual amount of software faults can be present in MM using multi-fragmentation approach [17]. Using this approach, the model can be divided into N fragments that are with the same structure but may differ in one or more parameter values [17]. The number of fragments N in the MM depends on the number of expected undetected software faults N_d, the value of which can be estimated using probabilistic prediction models [9]. Based on the assumption on equivalence of design faults failure rates for diverse software the general system design failure rate is

$$\lambda_{d1} = \lambda_{d2} \rightarrow \lambda_d = \lambda_{d1} + \lambda_{d2} \ . \tag{1}$$

b. $\mu_{d(i)}$ – design failure recovery rate. Using assumption of design recovery rates equivalence, the total system design recovery rate is design recovery rate is

$$\mu_{d1} = \mu_{d2} \rightarrow \mu_d = \mu_d / \left(\sum_{i=1}^{2} \frac{\lambda_{d(i)}}{\mu_{d(i)}} \right) . \tag{2}$$

c. $\lambda_{p(i,j)}$ – failure rate for the failures caused by physical faults in the track $T_{i,j}$, where $i \leq 2$, $j \leq 3$. As each track consist of five modules, the total $\lambda_{p(i,j)}$ of the track $T_{i,j}$ can be calculated as

$$\lambda_{p(i,j)} = \lambda_{DIM(i,j)} + \lambda_{DOM(i,j)} + \lambda_{LM(i,j)} + \lambda_{AIM(i,j)} + \lambda_{AOM(i,j)} \ , \tag{2}$$

where $\{\lambda_{DIM(i,j)}, \lambda_{DOM(i,j)}, \lambda_{LM(i,j)}, \lambda_{AIM(i,j)}, \lambda_{AOM(i,j)}\}$ – physical failure rates of DIM, DOM, LM, AIM, AOM, respectively. All corresponding components of the tracks are identical, their failure rates for the failures caused by physical faults are also equal. Thus, value $\lambda_{p(i,j)}$ is equal for all $T_{i,j}$.

d. $\mu_{p(i,j)}$ – recovery rate for the failures caused by physical faults in the track $T_{i,j}$. Equally to previous case the total recovery rate of the $T_{i,j}$ can be calculated using

$$\mu_{p(i,j)} = \lambda_{p(i,j)} / \left(\frac{\lambda_{DIM(i,j)}}{\mu_{DIM(i,j)}} + \frac{\lambda_{DOM(i,j)}}{\mu_{DOM(i,j)}} + \frac{\lambda_{LM(i,j)}}{\mu_{LM(i,j)}} + \frac{\lambda_{AIM(i,j)}}{\mu_{AIM(i,j)}} + \frac{\lambda_{AOM(i,j)}}{\mu_{AOM(i,j)}} \right) \quad (4)$$

where $\{\mu_{DIM(i,j)}, \mu_{DOM(i,j)}, \mu_{LM(i,j)}, \mu_{AIM(i,j)}, \mu_{AOM(i,j)}\}$ – physical failure recovery rates of DIM, DOM, LM, AIM, AOM, respectively. Using the same principle as for $\lambda_{p(i,j)}$, the $\mu_{p(i,j)}$ is equal for all system tracks.

e. D_i – the reliability of control and diagnostic tools, where $i \leq 2$. Table 2 contains the MM parameters values.

TABLE I
MARKOV MODEL PARAMETERS VALUES

Parameter	λ_d	μ_d	λ_p	μ_p	D
Value	10^{-5}	0.01	10^{-4}	1	{0.95, 0.99}

The basic fragment of MM is presented on Fig. 3.

The system operates as follows. At initial moment of time all channels provide non-stop correct functioning and system is in state $S_{0(3,3)}$, where (3, 3) shows that main channel operates on 3 tracks as well as the diverse channel. At random moment the physical failure occurs on one of the tracks in main or diverse channel, and if the failure was identified by self-diagnostic tool the system moves to the state $S_{1(3,2/2,3)}$ with rate $6\lambda_p D$ in other case system moves to the state $S_{1(3,2f/2f,3)}$ with rate $6\lambda_p(1-D)$.

The abbreviation (3,2/2,3) shows that two tracks are left in main or diverse channel and the failure was caught during diagnostic; abbreviation (3,2f/2f,3) presents situation when failure occurred in one of the channels but was not discovered by self-diagnostic tool. Such abbreviations were used to present states in basic fragment of MM.

If after the work of system maintenance group (operation in state $S_{1(3,2/2,3)}$) no new failure have occurred, the system recovers back into state $S_{0(3,3)}$ with rate μ_p. If the failure occurred on the same channel as in previous case and was identified during diagnostic the system moves to the state $S_{4(3,1/1,3)}$ with rate $2\lambda_{<}D$, and if failure was not detected – to the state $S_{5(3,1f/1f,3)}$ with rate $2\lambda_p(1-D)$. The system recovers to the state $S_{1(3,2/2,3)}$ from $S_{4(3,1/1,3)}$ with rate μ_p. The states $S_{4(3,1/1,3)}$ and $S_{5(3,1f/1f/3)}$ presents situation when main or diverse channel goes to the failed state. In case if failure occurs in the channel, which shows constant functioning on all three tracks and was caught by self-diagnostic tools, the system moves to the state $S_{3(2,2)}$ with rate $3\lambda_p D$, in opposite case – to the state $S_{6(2f,2/2,2f)}$ with rate $3\lambda_p(1- D)$. The system recovers to the state $S_{1(3,2/2,3)}$ from $S_{3(2,2)}$ with rate μ_p.

The same process of failure occurrence, detection and elimination continuous till states $S_{11(1,1)}$, $S_{12(1f,1/1,1f)}$, $S_{13(1f/1f)}$ which shows the total system failure.

The Fig. 4 shows the transition from first MM fragment to the second based on occurrence of design failures in each state which present at least one working channel. We did not present the internal transitions in each fragment to increase readability of the design failure detection and elimination processes.

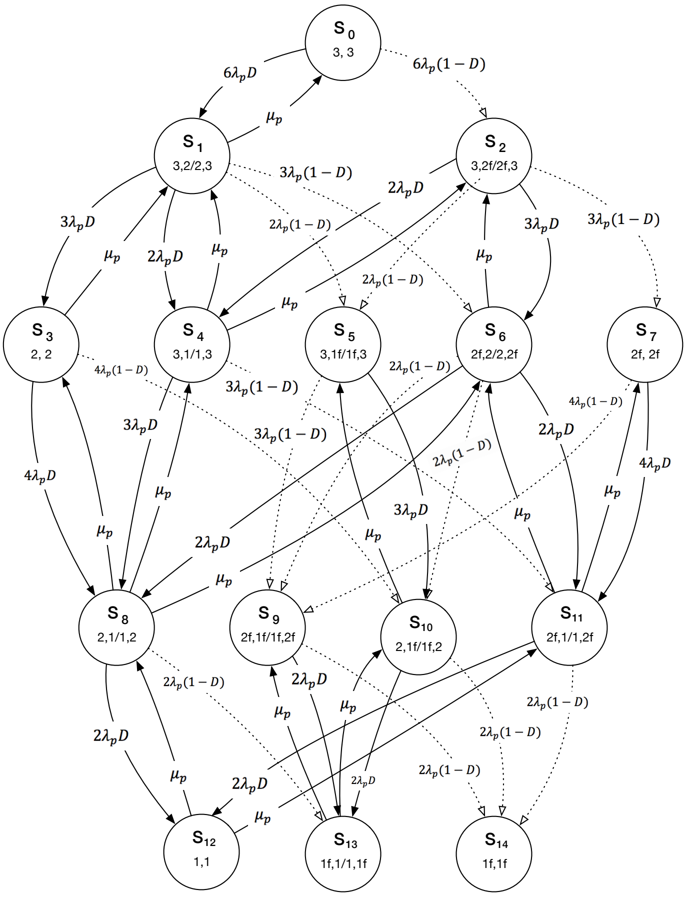

Fig. 3 – Basic fragment of Markov model

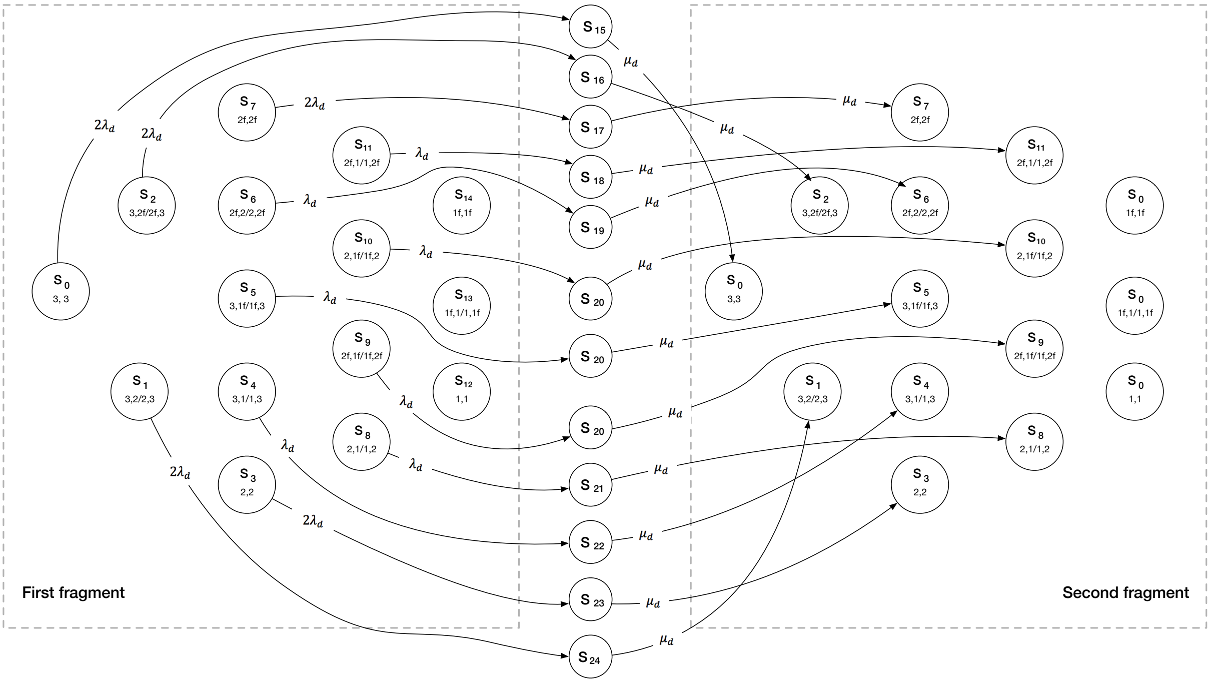

Fig. 4 – Design faults detection and elimination

IV. Safety Analysis Results

We have applied the metric-based approach to select the T&T for further MM analysis. Initially it was presented in [15] and aims to reduce the T&T selection risks, increase an accuracy and optimize time and computational resources, which are spend during assessments. During the use of metric-based approach four test are consequently applied on the given MM. Each test determines on scale [0, 1] the level of four MM features, namely stiffness, decomposability, sparsity and fragmentedness, where 0 points on absence of the feature and 1 shows that given feature is strongly presented in MM. As a result, the modeler gains the recommendation of which technique is preferable for MM analysis. According to the same approach we have applied multicriteria optimization to determine the tool, which accounts following preferences: operation system support, necessary internal functions, accuracy of implemented numerical methods and use of decimal data type.

MM features test showed that presented model is moderately stiff with 0.167 value, completely decomposable with 0.02, highly sparse with 0.2 and moderately fragmented with 0.3 value. As a result, we have got the recommendation to use stiffness-tolerance methods (implicit Runge-Kutta, TR-BDF2 [11] etc.).

The multicriteria optimization was applied under the following set of preferences:

- manual (graphical or analytical) MM construction;
- functions for internal stiffness and sparsity detection;
- implemented numerical methods with accuracy at least 10^{-5}; - availability on Windows and Linux operation systems; - presence of decimal data type.

As the result we have obtained recommendation for Mathematica and/or Matlab application. To make additional verification of obtained results with selected T&T we applied utility EXPMETH, that has been validated extensively on a range of models [13 – 15, 17]. The utility uses the algorithm of modified exponential method.

To analyze the RTS safety level we can assess an availability function *A(t)*, which can be calculated as the sum of system working states probabilities, with initial condition *A(0) = 1*

$$A(t) = \sum_{i=1}^{n} P_i(t), i \in N \quad , \tag{5}$$

where $P_i(t)$ is a probability of being in working state *i* at moment *t*.

The RTS working states are presented with the set $W = \{S_{0(3,3)}, S_{1(3,2/2,3)}, S_{2(3,2f/2f,3)}, S_{3(2,2)}, S_{4(3,1/1,3)}, S_{5(3,1f/1f,3)}, S_{6(2f,2/2,2f)}, S_{7(2f,2f)}, S_{8(2,1/1,2)}, S_{9(2f,1f/1f,2f)}, S_{10(2,1f/1f,2)}, S_{11(2f,1/1,2f)}\}$.

Fig. 5 presents the result of *A(t)* for case *D = 0.95* and result for D = 0.99 is shown on Fig. 6. Analysis was provided on the time interval $t \in [0; 10\,000]$ hours.

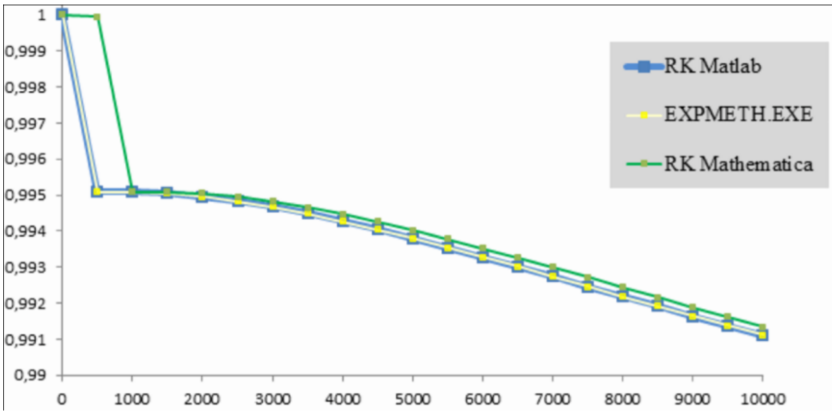

Fig. 5 RTS availability function for D = 0.95

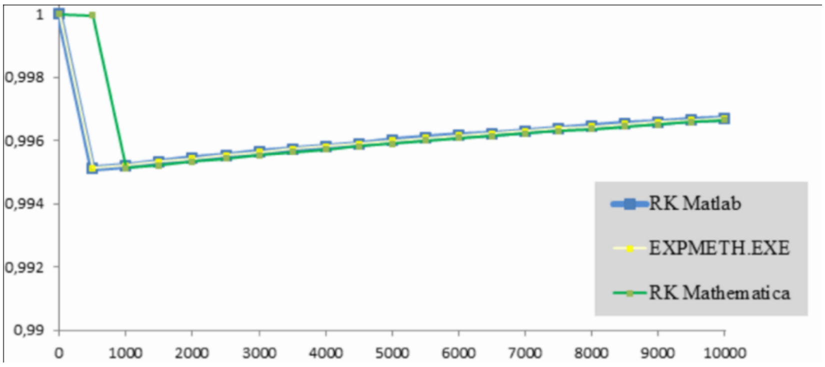

Fig. 6 RTS availability function for D = 0.99

The minimal and maximal difference between *A(t)* values obtained with Matlab, Mathematica, EXPETH tools for D = 0.95 and D = 0.99 are presented in Table 2.

TABLE II
DIFFERENCE BETWEEN A(T) VALUES

D	Min/Max	Matlab & EXPMETH	Matlab & Mathematica	EXPMETH & Mathematica
D = 0.95	Min	$4.7*10^{-9}$	$1.3*10^{-5}$	$1.3*10^{-5}$
	Max	$1.63*10^{-6}$	$4.85*10^{-3}$	$4.86*10^{-3}$
D=0.99	Min	$1.17*10^{-8}$	7*10 -5	$7.1*10^{-5}$
	Max	$1.74*10^{-6}$	$4.82*10^{-3}$	$4.82*10^{-3}$

V. CONCLUSION

The paper describes a case study of typical RTS architecture analysis and assessment of safety parameters using the classical state-space modelling approach – Markov modelling. The metric-based approach was applied to select T&T under the set of initial preferences. Based on the obtained values we can conclude that under all assumptions presented in Section 3 and with given initial parameters values the studied architecture constructed on FPGA-based digital platform, provides the safety level which can comply the “SIL 2” according to [2].

In our future work we intend to analyze the system behavior in case if the self-diagnostic tool could not detect the design faults.

REFERENCES

[1] M. Yastrebenetskiy et al., “Nuclear Power Plants Safety: Instrumentation and control systems”, Kiev: Osnova-Print, 2011, p. 768.

[2] IEC 61508, “Functional safety of electrical/electronic/programmable electronic safety-related systems”, IEC Standards, 2010, p. 594.

[3] S. Archana, R. Srinivasan, K. S. Trivedi. “Availability Models in Practice”, in Proc. Int. Workshop on FaultTolerant Control and Computing (FTCC-1), Seoul, 2000, pp. 1-24.

[4] ECSS-Q-ST-30-09: 2008, “European Cooperation for Space Standardization (ECSS): Availability analysis”, 2008.

[5] M. Malhotra, J. K. Muppala, K. S. Trivedi, “Stiffness-Tolerant Methods for Transient Analysis of Stiff Markov Chains”, Microelectronic Reliability, vol.34(11), 1994, pp.1825-1841.

[6] A. Reibman, K. S. Trivedi, “Numerical Transient Analysis of Markov models”, Comput. Opns. Res., vol.15(1), 1988, pp. 19-36.

[7] A. Bobbio, K. S. Trivedi, “A Aggregation Technique for Transient Analysis of Stiff Markov Chains”, IEEE Trans. on Comp., C-35, pp. 803-814, 1986.

[8] G. H. Golub, C. F. Van Loan, “Matrix Computations”, JHU Press, 1996, p. 694.

[9] J. Stoer, R. Bulirsch, “Introduction to Numerical Analysis”, Springer, 2002, p.732.

[10] A. Reibman, K. S. Trivedi, S. Kumar, and G. Ciardo, “Analysis of Stiff Markov Chains”, ORSA Journal on Computing, vol.1(2), pp.126-133, 1989.

[11] E. Hairer, G. Wanner, “Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems”, R. Bank, Ed. Springer, 2010, p. 631.

[12] IEC 60800, “Nuclear power plants – Instrumentation and control system important for safety – Software aspects for computer-based systems performing category A functions”, 2006.

[13] V. Kharchenko, O. Odarushchenko, V. Odarushchenko and P. Popov, “Availability Assessment of Computer Systems Described by Stiff Markov Chains: Case Study”, Springer, CCIS , vol. 412, 2013, pp. 112 - 135.

[14] V. Kharchenko, et. al., “Assessment of the Reactor Trip System Dependability: Two Markov’s Chains – Based Cases”, in Conf. Proc. Of the 10th Int. Conf. Digital Technologies, Zilina, 2014, pp. 103 – 109.

[15] V. Kharchenko et.al, “Metric-based approach and Tool for Modeling the NPP I&C System Using Markov Chains”, in Conf. Proc. of 23nd ICONE, vol.5, 2014.

[16] Littlewood, B., Popov, P., Strigini, L., “Modelling Software Design Diversity - a Review. ACM Computing Surveys”, vol.33 (1), 2001, pp. 177 – 208.

[17] V. Butenko, “Modeling of a Reactor Trip System Using Markov Chains: Case Study”, in Conf. Proc. of 22nd ICONE, vol. 5, 2014.

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