Availability Model of Critical NPP I&C Systems Considering Software Reliability Indices

Bogdan Volochiy, Vitaliy Yakovyna, Oleksandr Mulyak

National University Lviv Polytechnic, 12 Bandera St., 79013, Lviv, Ukraine

bvolochiy@ukr.net, yakovyna@polynet.lviv.ua,
mulyak.oleksandr@gmail.com

Abstract. Providing the high availability level for the Instrumentation and Control (I&C) Systems in Nuclear Power Plants (NPP) is highly important. The availability of the critical NPP I&C systems depends on the hardware and software reliability behavior. The high availability of the I&C systems is ensured by the following measures: structural redundancy with choice of the I&C system configurations (two comparable sub-systems in the I&C system, majority voting "2oo3", "2oo4", etc.); maintenance of the I&C system, which implies the repair (changing) of no operational modules; using the N-version programming; software updates; automatic software restart after temporary interrupts caused by the hardware fault. This paper proposes solution of the following case: the configuration of the fault-tolerant I&C system with known reliability indexes of hardware (failure rate and temporary failure rate) is chosen, the maintenance strategy of hardware (mean time to repair, numbers of repair), methods to forecast the number of software failures and the failure rate is specified. To solve this issue, the availability model of the fault-tolerant I&C system was developed in the discrete-continuous stochastic system form. We have estimated the influence of the I&C system on the operational software parameters. Two configurations of I&C systems are presented in this paper: two comparable subsystems in I&C system, and I&C system with majority voting "2oo3".

Keywords. Instrumentation and Control (I&C) System, Discrete-Continuous Stochastic System, Reliability Behavior, Structural-Automated Model, Markovian Chains, Software Reliability

Key Terms. Mathematical Modeling, Method, Software Systems

ICTERI 2016, Kyiv, Ukraine, June 21-24, 2016

1 Introduction

1.1 Motivation

Nowadays the development of fault-tolerant computer-based systems (FTCSs) is a part of weaponry components, space, aviation, energy and other critical systems. One of the main tasks is to provide requirements of reliability, availability and functional safety. Thus the two types of possible risks relate to the assessment of risk, and to ensuring their safety and security.

Reliability (dependability) related design (RRD) [1-6] is a main part of development of complex fault-tolerant systems based on computers, software (SW) and hardware (HW) components. The goal of RRD is to develop the structure of FTCS tolerating HW physical failure and SW designs faults and assure required values of reliability, availability and other dependability attributes. To ensure fault-tolerance software, two or more versions of software (developed by different developers, using other languages and technologies, etc) are used [7].Therefore use of structural redundancy for FTCS with multiple versions of software is mandatory. When commissioning software some bugs (design faults) remain in its code [8], this leads to the shutdown of the FTCS. After detection the bugs, a software update is carried out. These factors have influence on the availability of the FTCS and should be taken into account in the availability indexes. During the operation of FTCS it is also possible that the HW will fail leading to failure of the software. To recover the software operability, an automatic restart procedure, which is time consuming, is performed. The efficiency of fault-tolerant hardware of FTCS is provided by maintenance and repair.

Insufficient level of adequacy of the availability models of FTCS leads either to additional costs (while underestimating of the indexes), or to the risk of total failure (when inflating their values), namely accidents, material damage and even loss of life. Reliability and safety are assured by using (selection and development) fault-tolerant structures at RRD of the FTCS, and identifying and implementing strategies for maintenance. Adoption of wrong decisions at this stage leads to similar risks.

1.2 Related Works Analysis

Research papers, which focus on RRD, consider models of the FTCS. Most models are primarily developed to identify the impact of one the above-listed factors on reliability indexes. The rest of the factors are overlooked. Papers [4, 5] describe the reliability model of FTCS which illustrates separate HW and SW failures. Paper [6] offer reliability model of a fault-tolerant system, in which HW and SW failures are differentiated and after corrections in the program code the software failure rate is accounted for. Paper [8] describes the reliability model of the FTCS, which accounts for the software updates. In paper [10] the author outlines the relevance of the estimation of the reliability indexes of FTCS considering the failure of SW and recommends a method for their determination. Such reliability models of the FTCS produce analysis of its conditions under the failure of SW. This research suggests that $MTTF_{system}=MTTF_{software}$. Thus, it is possible to conclude that the author considers the HW of the FTCS as absolutely reliable. Such condition reduces the credibility of the result, especially when the reliability of the HW is commensurable to the reliability of the SW. Paper [11] presents the assessment of reliability parameters of FTCS through modeling behavior using Markovian chains, which account for multiple software updates. Nevertheless there was no evidence of the quantitative assessments of the reliability measures of presented FTCS.

In paper [12], the authors propose a model of FTCS using Macro-Markovian chains, where the software failure rate, duration of software verification, failure rate and repair rate of HW are accounted for. The presented method of Macro-Markovian chains modeling [12, 13] is based on logical analysis and cannot be used for profound configurations of FTCS due to their complexity and high probability of the occurrence of mistakes. Also there is a discussion around the definition of requirements for operational verification of software of the space system, together with the research model of the object for availability evaluation and scenarios preference. It is noted that over the last ten years out of 27% of space devices failures, which were fatal or such that restricted their use, 6% were associated with HW failure and 21% with SW failure.

Software updates are necessary due to the fact that at the point of SW commissioning they may contain a number of undetected faults, which can lead to critical failures of the FTCS. Presence of HW faults relates to the complexity of the system, and failure to conduct overall testing, as such testing is time consuming and needs substation financial support. To predict the number of SW faults at the time of its commissioning various models can be used, one for example is Jelinski-Moranda [14].

A goal of the paper is to suggest a technique to develop a Markovian chain for critical NPP I&C system with different redundancy types (first of all, structure and version) using the proposed formal procedure and tool. The main idea is to decrease risks of errors during development of Markovian chain (MC) for systems with very large (tens and hundreds) number of states. We propose a special notation which allows supporting development chain step by step and designing final MC using software tools. The paper is structured in the following way. The aim of this research is calculating the availability function of critical NPP I&C system with version-structural redundancy and double software updates.

To achieve this goal we propose a newly designed reliability model of critical NPP I&C system. As an example a special critical NPP I&C system is researched (Fig.1). The following factors are accounted for in this model: overall reserve of critical NPP I&C system and joint cold redundancy of modules of main and diverse systems of critical NPP I&C system; the existence of three software versions; SW double update; physicals fault.

Structure of the paper is the following. Researched critical NPP I&C system is described in the second section. An approach to developing mathematical model based on Markovian chain and detailed procedure for the critical NPP I&C system are suggested in the third and fourth sections correspondingly. Simulation results for researched Markov's model are analyzed in the section 4. Last section concludes the paper and presents some directions of future researches and developments.

2 Researched Typical NPP Instrumentation and Control Systems Based on Digital Platform

Here we provide the structure (Fig.1) of researched typical NPP Instrumentation and Control system (I&C) based on the digital platform [15]. This platform consists of main and diverse systems which are based on the Field Programmable Gates Arrays (FPGA) chips [16]. Main and Diverse systems based on the FPGA safety controller (FSC) with three parallel channels on voting logic “2-out-of-3”.

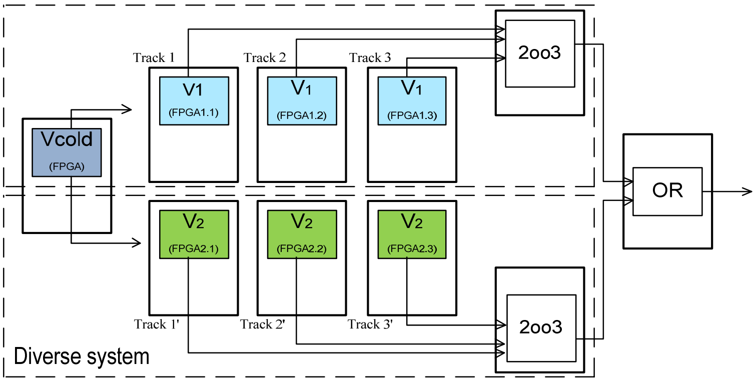

Fig. 1. Configuration of critical NPP I&C systems

This architecture consists of two system (main, diverse) each of them consists three channels connected in parallel with majority voting arrangement for the output signals, such that the output state is not charged if only one channel gives a different result, which disagrees with the other two channels.

The signals from Main and Diverse systems are comparing by element OR.

3 Methods to Forecast the Number of Software Failures and the Software Failure Rate

The papers [18, 19] describe methods of predicting numbers of undetected SW design faults. This method is based on the SW reliability model with index of complexity [20, 21]. The SW reliability model [20] describes the behavior of SW failures in non homogeneous Poisson process forms. The cumulative number of SW failures up to time *t* is calculated based on formula (1):

$$m(t) = \alpha\left(-\beta^s t^s e^{-\beta t} + sG_{\beta t}(s)\right), \tag{1}$$

where $G_z(p) = \int\limits_0^z t^{p-1} e^{-t} dt$ – an incomplete gamma function, α – the coefficient describes the total number of SW failure, β – the factor that represents the rate of detection of SW failures, s – an index of SW complexity.

Work [21] researches and specifies the intervals of value of the complexity index of SW ***s***. This circumstance has allowed for the elaboration of a formal selection rule for SW reliability models with different complexity indexes. The total number of SW failures (and, consequently, the total number of SW design faults N_{def}, on condition that one SW failure is caused by one SW design fault) is determined by the value of the function of the cumulative number of SW failures (1) at $\boldsymbol{t\to\infty}$**:**

$$N_{def} = m(\infty) = \alpha s G(s), \tag{2}$$

where G(s) – the Gamma function.

To estimate the undetected numbers of SW design faults, the following steps [22] should be performed:

- carry out SW testing and represent the result as the number of SW failures in defined interval. The input range of statistical sampling is divided into equal interval $l \leq 5\lg(n)$ (where n – the total number of SW failures obtained during testing);
- define the point estimates of the reliability SW model parameters α, β, and define parameter s by using the method of maximum likelihood [20];
- carry out the Kolmogorov – Smirnov test for quality, reviewing the experimental reliability model described;
- use the point estimates of the reliability SW model parameters according to (2) the defined total number of SW design faults Ndef . The forecast for the number of undetected SW design faults is obtained by subtracting the total number of SW design faults Ndef and defined SW design faults.

Using regression analysis [18, 19], it is possible to:

- increase the accuracy of the forecast of the total number of SW design faults using formula (2); or
- decrease the time required to forecast the number of SW design faults.

The number SW faults depends on the duration of SW testing, which provides information about SW failure behavior. The variable N_{def} from formula (2) was estimated using a nonlinear regression with explanatory variables T_i – time of SW testing. The following equation (3) was used as the regression equation.

$$N_{def}(T_i) = A\left(1 - \exp\left(-\left(k(T_i - T_c)\right)^d\right)\right), \tag{3}$$

where A, k, d, T_c – parameters of the regression equation.

It is then possible to determine the adjusted forecast of the total number of SW failures N^*_{def} from equation (3) on condition of the time of SW testing being unlimited ($T_i \to \infty$). Based on the equation (3) the total number of SW failure is equal to the value of regression parameter $\boldsymbol{A}$.

To estimate the adjusted forecast for the total number of undetected SW failures N^*_{def}, the following steps should be performed:

- during the SW testing procedure, it is necessary to calculate the point estimates of the reliability SW model parameters **α, β** and ***s*** [20] by using the methods of maximum likelihood on the interval (0; T_i), where T_i - the current moment of SW testing. It is also is necessary to calculate $N_{def}(T_i)$ according the equation (2);
- estimate the parameter of regression equation (3) by using the least squares method for set of $N_{def}(T_i)$; $N^*_{def} = A$;
- the forecast number of undetected SW faults is determined by subtracting the number of detected and fixed SW faults from N^*_{def};
- in the case where a continuation of the process for SW testing is necessary, go back to step 1 and continue adding the new value to set $N_{def}(T_i)$.

An example of dependence $N_{def}(T_i)$ [19] which was obtained during the SW testing procedure is presented in figure 2.

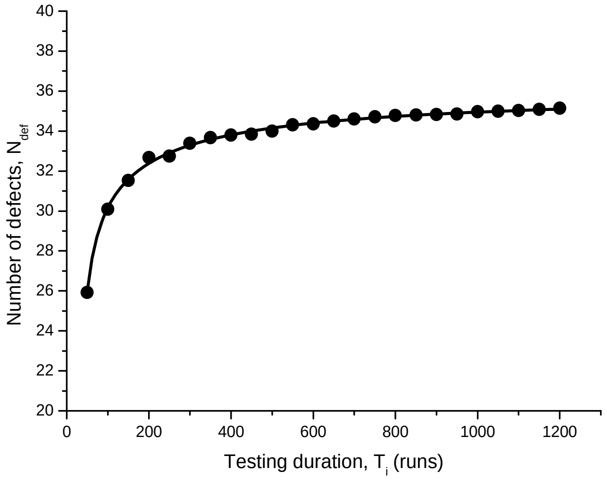

Fig. 2. Dependents of forecasting the numbers of SW faults N_{def} (points), which was calculated according equation (2) from the SW testing durations and appropriated regression equation (line)

In this case, using the methods of forecasting t the SW failure numbers and equation (3) increases the accuracy of forecasting by 2-3%. Also, this method decreases the time required to forecast the number of SW failures [19].

An advantage of the SW reliability model [20] is that it is possibile to estimate the SW failure rate based on SW testing results at the appropriate level of the life cycle. The SW failure rate depends on the time of SW testing (this dependence is caused by correction of the SW faults on the appropriate live cycle). The relationship takes the form (4):

$$\lambda(t) = \frac{dm(t)}{dt} = \alpha\beta^{s+1}t^s e^{-\beta t}, \tag{4}$$

As a result of using equation (4), the point value of the model parameters and the duration of SW , it is possible to calculate the SW failure rate λ_{SW} – which is constant in time. It is necessary to estimate the value of λ_{SW}, the availability of the I&C NPP system based on Markovian analysis.

The authenticity of the estimate of the undetected SW faults [23, 24] is provided by forecasting the SW failure numbers (as result SW faults) based on artificial neural networks (NN) with radial basis function (RBF). The NN RBF is a nonparametric model of behavior of SW reliability which does not require a priori knowledge and assumptions about the behavior of SW failure. In this research, input data about SW failures were presented in cumulative time series form. The cumulative time series is used for learning about the neural networks RBF and for forecasting the value of SW failure on subsequent time series.

The most reasonable results of forecasting SW failure are obtained by using NN RBF with an Inverse Multi-quadratic function (10 neurons in input layer and 30 neurons hidden layer) [24]. In this configuration, the mean square error of approximation is 1,0%. The coefficient of determination between the forecasting and controlled series is 0,9965. Although the accuracy of forecasting decreases by 1,7%, it is possible to reduce the duration of learning time of the neural network by 3-6 times by using a Gaussian function (15 neurons in the input layer and 10 neurons in the hidden layer) [23, 24].

As a result of the different SW systems analysed, a configuration of neural network RBF was conducted that could be used for time series forecast with homogeneous failure process represented by a cumulative time series.

Figure 3 presents an example of forecasting t, specifically, the total number of SW failures of the web-browser Chromium forecast using the neural network RBF with parameters listed above.

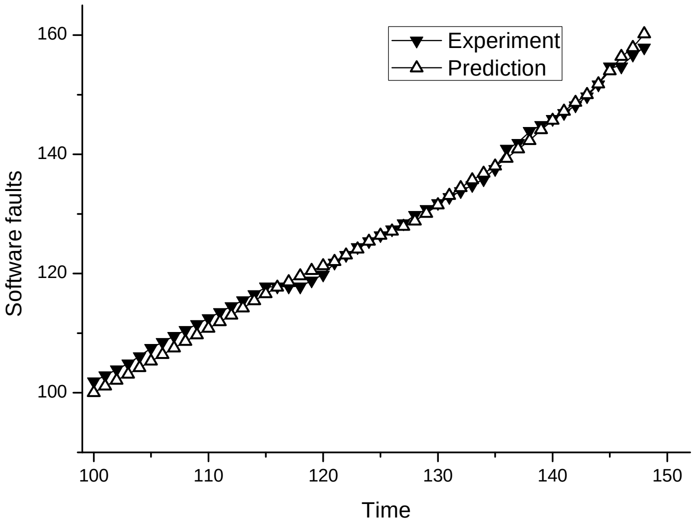

Fig. 3. An example of forecasting t, the total number of SW failures of web-browser Chromium, using the neural network RBF

This parts of paper outlines the estimated numbers of undetected SW faults using two methods based on regressions analysis and neural networks. This is used for reliably estimating the number of undetected SW faults and ensures the requirements of standard [25] are satisfied. It is considered acceptable when number of SW faults calculated by two methods is equal to or less than the standards requirement.

4 Markov's Model for Critical NPP I&C Systems with Software Updates

The method of automated development the Markovian chain of the researched critical NPP I&C systems is described in the works [9, 26]. It involves a formalized representation of the object of study as a "structural-automated model". To develop this availability model of the critical NPP I&C systems one needs to perform the following tasks: develop a verbal description of the research object (fig. 1); define the basic events; define the components of vector states, which can be described as a state of random time; define the parameters for the object of research, which should be in the model; and shape the tree of the modification of the rules and component of the vector of states.

4.1 The Procedures to Describe Behavior of the Critical NPP I&C Systems

The critical NPP I&C systems behavior is described by the following procedures:

- *Procedure 1.* Detection the failure in the critical NPP I&C systems (hardware failure, software failure). Failure can occur in the Main system (MS) and Diverse system (DS).
- *Procedure 2.* Detection of failure in the MS or in the DS of the critical NPP I&C systems.
- *Procedure 3.* Connection of the module from cold standby to faulty systems.
- *Procedure 4.* Loading the software on the module with connections from cold standby to faulty systems.
- *Procedure 5.* Software updating.
- *Procedure 6.* Repair (replacement) of the HW of the faulty systems.

4.2 A Set of the Events for the Critical NPP I&C Systems

According to described procedures which determine the behavior of critical NPP I&C systems, a list of events is composed. Events are presented in pairs corresponding to the start and the end of time intervals to perform each procedure. From this list of events for "structural-automated model" basic events are selected [9].

As a result of analysis, seven basic events in particular were determined: Event 1 - "Hardware failure of the MS module"; Event 2 - "Software failure of the MS module"; Event 3 - "Hardware failure of the DS module"; Event 4 - "Software failure of the DS module"; Event 5 - "Completing of the module switching from cold standby to non-operational systems"; Event 6 - "Completing of the software updates procedure"; Event 7 - "Completing of the procedure of the hardware repair"

4.3 Components of Vector States for the Critical NPP I&C Systems

Components of the vector state that can also be described as a state of random time. To describe the state of the system, eleven components are used: V1 – displays the current number of modules in the MS (the initial value of components V1 equal to n); V2 – displays the current number of modules in the DS (the initial value of components V2 equal to k); V3 – displays the current number of modules in cold standby (the initial value of components V3 equal to m_c); V4 – displays which software version is operated by the MS (V4=0 – first version, V4=1 – second version, V4=2 – third version); V5 – displays which software version operated by DS (V5=0 – first version, V5=1 – second version, V5=2 – third version); V6 – displays the SW faults in the MS; V7 – displays the SW faults in the DS; V8 – displays the SW failure in the MS; V9 – displays the SW failure in the DS; V10 – displays the number of non-operational module, due to HW failure.

4.4 The Parameters of the Critical NPP I&C Systems Markov's Model

Developing Markov's model of the critical NPP I&C systems, its composition and separate components should be set to relevant parameters in particular: n – number of modules that are the part of the MS; k – number of modules that are the part of the DS; m_c –number of the modules in the cold standby;λ_{hw}– the failure rate that is in MS or DS and in the hot standby; λ_{sw11}, λ_{sw12} – the failure rate of first and second software versions;T_{up1}, T_{up2} – mean time of the first and second software updates; T_{switch} – mean time of the module connections from standby; T_{rep}– mean time of hardware repair.

4.5 Structural-Automaded Model of the Critical NPP I&C System for the Automated Development the Markovian Chain with Software Updates

According to the technology of a modeling, the discrete-continuous stochastic systems [9] based on certain events using the component vector state and the parameters that describe critical NPP I&C systems, and model of the critical NPP I&C systems for automated development of the Markovian chains are presented on the table 1. Below is describes the procedures of structural-automated model development:

Table 1. Structural-Automated Model of the critical NPP I&C systems for the automated development of the Markovian chains

Terms and conditions	Formula used for the intensity of the events	Rule of modification component for the state vector
Event 1. Hardware failure of the MS module		
(V1>=(n-1)) AND (V6=0)	$V1 \cdot \lambda_{hw}$	V1:=V1-1; V8:=V8+1
Event 2. Software failure of the MS module		
(V1>=(n-1)) AND (V4=0) AND (V6=0)	$V1 \cdot \lambda_{sw11}$	V1:=V1-1; V4:=0; V6:=1

Terms and conditions	**Formula used for the intensity of the events**	**Rule of modification component for the state vector**
(V1>=(n-1)) AND (V4=1) AND (V6=0)	$V1\cdot\lambda_{sw12}$	V1:=V1-1; V4:=1; V6:=1
Event 3. Hardware failure of the DS module		
(V2>=(k-1)) AND (V7=0)	$V2\cdot\lambda_{hw}$	V2:=V2-1; V8:=V8+1
Event 4. Software failure of the DS module		
(V2>=(k-1)) AND (V5=0) AND (V7=0)	$V2\cdot\lambda_{sw11}$	V2:=V2-1; V5:=0; V7:=1
(V2>=(k-1)) AND (V5=1) AND (V7=0)	$V2\cdot\lambda_{sw12}$	V2:=V2-1; V5:=1; V7:=1
Event 5. Completing of the module switching procedure from cold standby to non-operational systems		
(V1<(n-1)) AND (V3>0) AND (V8>0)	$1/T_{switch}$	V1:=V1+1; V3:=V3-1
(V2<(n-1)) AND (V3>0) AND (V8>0)	$1/T_{switch}$	V2:=V2+1; V3:=V3-1
Event 6. Completing of the software updates procedure		
(V1<n) AND (V4=0) AND (V6=1)	$1/T_{up1}$	V1:=n; V4:=1; V6:=0
(V1<n) AND (V4=1) AND (V6=1)	$1/T_{up2}$	V1:=n; V4:=2; V6:=0
(V2<k) AND (V5=0) AND (V7=1)	$1/T_{up1}$	V2:=k; V5:=1; V7:=0
(V2<k) AND (V5=1) AND (V7=1)	$1/T_{up2}$	V2:=k; V5:=2; V7:=0
Event 7. Completing of the procedure of the hardware repair		
(V1<n) AND (V2=k) AND (V6=0) AND (V8>0)	$1/T_{rep}$	V1:=n; V6:=0; V8:=0
(V1=n) AND (V2<k) AND (V7=0) AND (V8>0)	$1/T_{rep}$	V2:=k; V7:=0; V8:=0
(V1<n) AND (V2<k) AND (V6=0) AND (V7=0) AND (V8>0)	$1/T_{rep}$	V1:=n; V2:=k; V8:=0

The number of software updates can be also changed. It is necessary to change vectors V4 and V5 the *event 6*, that are responsible for the number of updates. For example, if there are three software updates, the entry component of the event will be as follows:

(V1<n) AND **(V4=2)** AND (V6=1)	**$1/T_{up3}$**	V1:=n; **V4:=3;** V6:=0
(V2<k) AND **(V5=2)** AND (V7=1)	**$1/T_{up3}$**	V2:=k; **V5:=3;** V7:=0

4.6 Automated Development of the Markovian Chain and Determining of Availability Function

The developed availability model of the critical NPP I&C system gives the possibilities according to technology [9] for automated construct of the Markovian chains. This construction provides a software module ASNA [17]. The Markovian chains which take into account the following settings critical NPP I&C system: n=3; k=3; m_c=0; λ_{hw}; λ_{sw11}, λ_{sw12}; T_{up1}, T_{up2}; T_{switch};T_{rep}, are consists of 169 state and 436 transitions. Information is available on the status of each software module ASNA we have on file "vector.vs", which is written in the form:

State 1: V1=3; V2=3; V3=0; V4=0; V5=0; V6=0; V7=0; V8=0
State 2: V1=2; V2=3; V3=0; V4=0; V5=0; V6=0; V7=0; V8=1
State 3: V1=1; V2=3; V3=0; V4=0; V5=0; V6=0; V7=0; V8=2
……….
State 169: V1=1; V2=1; V3=0; V4=2; V5=2; V6=0; V7=0; V8=4

As the configurations of researched critical NPP I&C system changes the dimension of graphs increases. Therefore for the configuration of critical NPP I&C sys (Fig. 1) with one module in a cold standby graph has 506 states and 1434 transitions.

The proposed availability model of critical NPP I&C system can be easily transformed for other features of the object of study. It is enough to: add / remove basic event; attach / remove components of the state vector; and include / exclude parameters that describe the studied system. Based on information about the work of critical NPP I&C system an appropriate change in the model could be made (Fig. 1).

Basing on the Markovian chains formulas for designing of availability critical NPP I&C system can be assembled. One measure of the availability of recovered critical NPP I&C system reveals it is an availability function. Availability functions of critical NPP I&C system is calculated as the sum of the probability functions staying in operable states of chains. Basing on these states the critical NPP I&C system availability function with parameters of critical NPP I&C is determined by the formula (5):

$$\begin{aligned}A(t)={}&\sum_{i=1}^{10}P_i(t)+\sum_{i=12}^{14}P_i(t)+\sum_{i=16}^{18}P_i(t)+\sum_{i=21}^{22}P_i(t)+\sum_{i=26}^{51}P_i(t)+P_{53}(t)+\sum_{i=55}^{61}P_i(t)+\\&+\sum_{i=67}^{72}P_i(t)+P_{74}(t)+\sum_{i=76}^{77}P_i(t)+P_{79}(t)+\sum_{i=81}^{83}P_i(t)+\sum_{i=86}^{87}P_i+\sum_{i=91}^{95}P_i+\sum_{i=101}^{146}P_i+\\&+P_{148}(t)+\sum_{i=150}^{151}P_i+P_{153}(t)+\sum_{i=155}^{158}P_i+P_{160}(t)+\sum_{i=162}^{163}P_i+P_{165}(t)+\sum_{i=167}^{168}P_i\end{aligned},\qquad(5)$$

Based on the Markovian chains ("vector.vs") a system of differential equations (6) was formed. Its solution allows us to estimate the function availability value of researched critical NPP I&C system.

$$\begin{aligned}\frac{dP_1(t)}{dt}&=-6\cdot(\lambda_{hw}+\lambda_{sw11})\cdot P_2(t)+\frac{1}{T_{repl}}\cdot(P_2(t)+P_3(t)+P_6(t)+P_7(t)+\\&\quad+P_8(t)+P_9(t)+P_{11}(t)+P_{16}(t))\\\frac{dP_2(t)}{dt}&=-\frac{1}{T_{repl}}\cdot P_2(t)-2\cdot\lambda_{sw11}P_2(t)-2\cdot\lambda_{hw}\cdot P_2(t)-3\cdot\lambda_{hw}\cdot P_2(t)+\\&\quad+2\cdot\lambda_{hw}P_1(t)\\\frac{dP_3(t)}{dt}&=-\frac{1}{T_{repl}}\cdot P_3(t)-3\cdot(\lambda_{hw}+\lambda_{sw11})\cdot P_3(t)+2\cdot\lambda_{hw}P_2(t)\\&\vdots\\\frac{dP_{169}(t)}{dt}&=-\frac{1}{T_{repl}}\cdot P_{169}(t)+2\cdot\lambda_{hw}P_{156}(t)+2\cdot\lambda_{sw12}P_{168}(t)\end{aligned},\qquad(6)$$

Initial conditions for the system (2) are: $P_1(t)=1; P_2(t)\ \ldots P_{169}(t)=0$.

5 Simulation Results

5.1 Research of Influence of Software Updates Duration on the Availability Function

With the assistance of the proposed model, the following questions can be answered: What are the duration values of the first and the second software update (ensuring the values of the availability function of critical NPP I&C system of the initial phase of its operation do not reach below the specified level)? What are the allowed duration values of the first and the second SW updates? How does the correlation between the first and the second SW updates influence on the availability function?

The experiment is conducted for the condition where the duration of the first software update is significantly shorter than the duration of the second update. The duration of the first update is given within 10 - 50 hours, and the duration of the second update - 200 hours. The experiment is conducted with the following parameters critical NPP I&C system: λ_{hw} = $1\cdot10^{-5}$ hour^{-1}; λ_{sw11} = $2\cdot10^{-3}$ hour^{-1}, λ_{sw12}=$1\cdot10^{-3}$ hour^{-1}; T_{switch}=6 min; T_{rep}=200 hour; T_{up2}=200 hour; (*line 1* -T_{up1}=10 hour; *line2* -T_{up1}=20 hour; *line3* -T_{up1}=30 hour; *line4* -T_{up1}=40 hour; *line5* -T_{up1}=50hour).

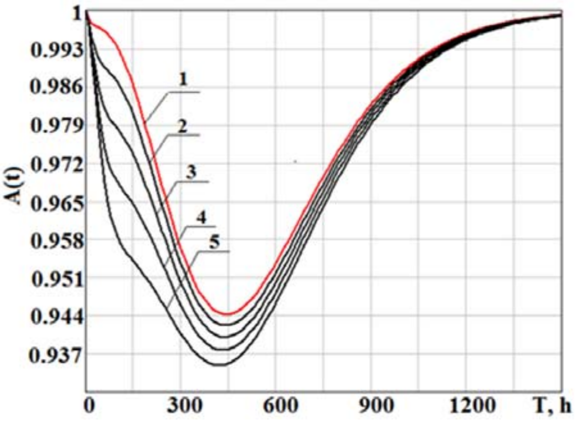

Fig. 4. Dependencies of availability function of the critical NPP I&C system on values of the software update durations (duration of the first software update for 10 to 50 hours; the duration of the second firmware update - 200 hours)

The following results are produced by the proposed experiments:

- The minimal decrease level of the availability function of the readiness of critical NPP I&C system in the first and the second experiments is the different. Hence could be argued that the first and second software updates has different influence on the reliability behavior of the critical NPP I&C system.
- With the assistance of the proposed model it is possible to choose the duration of software updates that helps to ensure a minimum allowed level of the decrease of the availability function of the critical NPP I&C system.

6 Conclusion

This research presents a model of critical NPP I&C system with double software updates to illustrate automated development of Markovian chains using a special technology and tool ASNA. Also this research presents two methods of forecasting the number of software failure with indexes of complexity and software failure rates.

The presented model can be easily adapted to different configurations of critical NPP I&C system, which envisages the use different majority voting, standby of the hardware part and as a consequence in the majority of software versions from different developers. In fact, this model can be adopted for an arbitrary number of software updates.

Future research has the potential to supplement this model with further factors:

- Erlang distribution for durations of software updates;
- Unsuccessful restarting; unreliable commutation of elements and so on.

References

1. Mudry, P.A., Vannel, F., Tempesti, G., Mange, D.: A reconfigurable hardware platform for prototyping cellular architectures. In: International Parallel and Distributed Processing Symposium. IEEE International, pp. 96--103 (2007)
2. Viktorov, O.: Reconfigurable Multiprocessor System Reliability Estimation. Asian Jounal of Information Technology 6 (9), pp. 958--960 (2007)
3. Rajesh, S., Vinoth Kumar C., Srivatsan, R., Harini, S., Shanthi, A.: Fault Tolerance in Multicore Processors With Reconfigurable Hardware Unit. In: 15thInternational conference on high performance computing. Bangalore, INDIA, pp. 166--171 (2008)
4. Amerijckx, C., Legat, J.-D.: A Low-Power Multiprocessor ArchitectureFor Embedded Reconfigurable Systems. In: Power and Timing Modeling, Optimization and Simulation, International Workshop, pp. 83--93 (2008)
5. Zhu, C., Gu, Z., Dick, R., Shang, L.: Reliable multiprocessor system-on-chip synthesis. In: Proc. International Conference Hardware/Software Co-design and System Synthesis, pp. 239--244 (2007)
6. Gostelow, K. P.: The design of a fault-tolerant, realtime, multi-core computer system. In: Proc. Aerospace Conference, IEEE, pp. 1--8(2011)
7. Lyu, M.R. (ed.): Software Fault Tolerance, John Wiley & Sons, New York, NY, USA (1995)
8. Korotun, T.M.: Models and methods for testing software systems. Programming problems, 2, 76--84 (2007) (in Russian)
9. Volochii, B.: Technology of modeling the information systems. Publishing NU "Lviv Polytechnic", Lviv, Ukraine (2004) (in Ukrainian)
10. Xiong, L., Tan, Q., Xu, J.: Effects of Soft Error to System Reliability. In: Proc. Workshops of International Conference on Advanced Information Networking and Applications. pp. 204--209 (2011)
11. Ponochonvyi, J.L., Odarushchenko, E.B.: The reliability modeling non-redundant information and control systems with software updated. Radio-electronic and Computer Systems, 4(8), 93--97 (2004) (in Russian)
12. Kharchenko, V., Odarushchenko, O., Odarushchenko, V., Popov, P.: Selecting Mathematical Software for Dependability Assessment of Computer Systems Described by Stiff Markov Chains. In: Ermolayev, V. et al. (Eds.): Proc. Int. Conf. ICTERI 2013, pp. 146--162 (2013)
13. Kharchenko, V., Ponochovny, Y., Boyarchuk, A.: Availability Assessment of Information and Control Systems with Online Software Update and Verification. In: Information and Communication Technologies in Education, Research, and Industrial Applications Communications in Computer and Information Science, Vol. 469, , pp. 300—324, Springer International Publishing, Switzerland (2014)
14. Moranda, P. B.: An error detection model for application during software development. IEEE Trans. Reliability, N. 4, 309--312 (1981)
15. Kharchenko, V., Sklyar, V., Volkoviy, A.: Development and Verification of Dependable Multi-Version Systems on the Basic of IP-Cores. In: Proc. Int. Conf. Dependability of Computer Systems (2008)
16. Review Guidelines for Field-Programmable Gate Arrays in Nuclear Power Plant Safety Systems. NUREG/CR-7006, U.S. Nuclear Regulatory Commission, Washington, D.C., USA (2010)
17. Bobalo, J., Volochiy, B., Lozynskyi, O., Mandzii, B., Ozirkovskyi, L., Fedasuk, D.,Shcherbovskyh, S., Jakovyna, V.: Mathematical models and methods for reliability

analysis of electronic, electrical and software systems, Lviv Polytechnic Press, Lviv, Ukraine (2013)

18. Yakovyna, V.S., Seniv, M.M., Harangda, I. J.: Improved procedure for determining the number of software defects in the early stages of testing. In: International scientific conference "Intelligent Decision Support Systems and Computational Intelligence problems" (ISDMCI'2012), pp. 238, Ukraine (2012)
19. Yakovyna, V.S., Fedasuk, D.V.: Improved procedure forecasting of software failures based on the reliability model with index of complexity. Software Engineering, 2, 5--13 (2012)
20. Chabanjuk, J.M., Yakovyna, V.S., Fedasuk, D.V.: Building and research the software reliability model with index value of the project. Software Engineering, 1, 24--29 (2010)
21. Yakovyna, V.S.: Modeling of software failure flow parameter and determine the series of complexity index. Proceedings of the National University "Lviv Polytechnic", 806, 296--302 (2014)
22. Yakovyna, V.S., Fedasuk, D.V., Seniv, M.M., Chabanjuk, J.M.: The method of estimation and forecasting software reliability based model with a dynamic index value of the project. Computing, 2(10), 97--107.
23. Yakovyna, V. S.: Influence of RBF neural network input layer parameters on software reliability prediction. In: Proc. 4-th International Conference Inductive Modelling (ICIM'2013). pp. 344--347, Kyiv (2013)
24. Yakovyna, V. S.: Software failures prediction using RBF neural network. Proceedings of Odessa Politechic University, 2(46), 111--118 (2015)
25. Nuclear power plants - Instrumentation and control important to safety - General requirements for systems. IEC 61513. (2011)
26. Volochiy, B., Mulyak, O., Ozirkovskyi, L., Kharchenko, V.: Automation of Quantitative Requirements Determination to Software Reliability of Safety Critical NPP I&C systems. In: Proc. 2nd Int. Symp. on Stochastic Models in Reliability Engineering, Life Science and Operations Management (SMRLO'16), Beer Sheva, Israel, February 15-18, pp. 337--346, IEEE CPS, 978-1-4673-9941-8/16, DOI 10.1109/SMRLO.2016.65 (2016)