UDC 004.93

ALGORITHMS AND SOFTWARE SUITE FOR RELIABILITY ASSESSMENT OF COMPLEX TECHNICAL SYSTEMS

Yakovyna V. S.[1, 2] – [1] Dr. Sc., Professor of Artificial Intelligence Department, Lviv Polytechnic National University, Lviv, Ukraine.

[2] Faculty of Mathematics and Computer Science, University of Warmia and Mazury in Olsztyn, Poland.

Seniv M. M. – PhD, Associate Professor of Software Department, Lviv Polytechnic National University, Lviv, Ukraine.

Symets I. I. – Assistant of Software Department, Lviv Polytechnic National University, Lviv, Ukraine.

Sambir N. B. – Assistant of Software Department, Lviv Polytechnic National University, Lviv, Ukraine.

ABSTRACT

Context. One of the most essential properties of technical systems is their reliability, i.e. the ability of the system to perform intended functions, preserving with time the values of operation indicators within the predefined boundaries. The failure cost for modern complex technical system can be very high, which can result in events of different severity ranging from economic losses to harm to human life and health. Hence, the requirements for their reliability constantly increase. The reliability assessment of complex technical systems can be simplified by the combination of analytical research methods with computational capabilities of modern computers. The most widely used analytical methods are based on the theory of Markov processes which in turn provide the possibility to determine the time dependencies of probabilities of the system to be in defined states (operating, recovering, failure etc.), and thus the values and time dependencies of the reliability indices needed. These methods can be successfully used for the reliability analysis of different kinds of technical systems: both non-recovered and recovered; non-redundant and redundant of different redundancy types, maintenance priorities etc. However, the application of these methods for complex technical systems containing large number of elements meets the high dimensional calculation problem, which makes it impossible to perform these tasks manually. Hence the problem of automation of complex technical system reliability modeling using modern computational systems is very relevant research topic. To solve this problem, one can use specific algorithmic and software techniques described in this paper.

Objective. The goal of this article is to develop the algorithms for automated RBD processing and reliability indices assessment of complex technical systems along with the software suite for automated reliability assessment.

Method. To perform the reliability analysis the RBD approach is used which allows one to represent and visualize each element of the system in the form of a rectangle, joined by the lines in parallel or in series with other elements of the system. To obtain the reliability indices values the mathematical model of technical system reliability behavior using Markovian random process was suggested. The algorithm of RBD processing and automatic determination of operability conditions of a technical system was further considered. To calculate the minimum and maximum number of operational and failure states for the system of n elements and r recoveries the paper introduces a mathematical model based on combinatorial approach. To develop the software suite the object-oriented approach was used.

Results. The algorithms and software suite allows us to easily construct RBD for a technical system, to automatically determine the operability condition with execution time of about 10 sec for 1,000 elements with mixed type of connection, to form automatically a state-and-transition matrix along with the corresponding differential equation system and solve it with total execution time of about 35 sec for 109 states and, thus to obtain the numerical values of reliability indices for the technical system studied. A case study of the reliability assessment for the system consisting of 22 elements using RBD shows that the total time of software execution is 36.712 sec. During executing of this test case the most time (35.168 sec) was spent for execution of the algorithm for construction of a state-and-transition graph consisting of 52,694 states.

Conclusions. The algorithms and methods for automated reliability indices assessment of complex technical systems based on RBD approach, as well as model for estimating the number of total and working system states are presented. The modular structure of the developed software suite makes it flexible and gives an opportunity to add and make modifications of modules fast and without significant program changes.

KEYWORDS: software, reliability, RBD, states and transitions graph.

ABBREVIATIONS

RBD is a reliability block diagram.

NOMENCLATURE

$P_i(t)$ is the probability of a system being at time t in the state x_i;

t is time;

x_i is a system state;

$\lambda_{ij}(t)$ is an intensity of transition from state x_i to state x_j;

N is the number of possible states of a system;

n is the number of elements (modules) in a system;

r is the number of recoveries for each element in a system.

INTRODUCTION

Development of complex technical systems sets new and still more complicated issues before design engineers. The majority of modern technical systems are hardware-software, multicomponent and thus difficult in engineering and development. Besides they are classified as restorable and unrestorable, of long and short operation time, standby and simplex, which also complicates the process of their design. One of the most important devel-

DOI 10.15588/1607-3274-2020-4-16

opment tasks especially in designing of such systems is the calculation of their reliability because the lives of dozens or even thousands of people depend on the operational activity of the above mentioned systems, their breakdown can lead to millions of losses and numerous deaths of people [1, 2]. Therefore, reliable design is one of the most important stages of technical system development in general and complex high-duty technical systems in particular.

The object of the study is the process of reliability assessment of complex technical systems.

The subject of the study is algorithms and software tools for automated reliability assessment using reliability block diagram.

The purpose of the work is to develop the algorithms and software suite for automated reliability assessment of complex technical systems using the RBD approach and Markov processes formalism.

1 PROBLEM STATEMENT

The main problem of designing of reliability models of complex technical systems based on Markovian processes is their large dimension, which reach hundreds of thousands of states even for a system consisting of relatively small number of elements, in particular if one or more of the following conditions are satisfied: (i) the elements are of different types; (ii) the elements can be recovered with a limited number of times; (iii) there are different maintenance teams and the teams are not universal (i.e. a particular team can repair only particular element types)[3].

Reliability behavior of a system in time can be interpreted as a random process with a discrete range of values (states) and continuous change of parameter (time), i.e. as a discrete-continuous stochastic process. Time dependences of the probabilities of random process can be described by the system of Kolmogorov – Chapman differential equations of the following type [4]:

$$\frac{dP_i(t)}{d(t)} = -\sum_{i=1}^{N} \lambda_{ij}(t) P_i(t) + \sum_{j=1}^{N} \lambda_{ij}(t) P_j(t), \qquad (1)$$
$$i,j = 1,2,\ldots, N.$$

A set of states and system transitions as well as the values of transition intensities (i.e. the failure and recovery intensities) are the input data for Kolmogorov–Chapman differential equations (1). These values are partially obtained directly from the user's input (failure and recovery intensities) and partially are formed automatically by the developed software suite based on a RBD provided by user [5].

The solution of equation system (1) allows us to determine the following reliability indices: probability of failure-free operation, the availability coefficient and availability function, the average failure-free operating time (Time-To-Failure), the average recovery time, etc. Thus, these values and theirs time dependencies are the output of the discussed software suite.

2 REVIEW OF THE LITERATURE

Reliability assessment of complex technical systems, including both purely hardware and with a software component, is often carried out using Markovian models. The essence of these models is in considering the system reliability behavior as a random process which can be described as a set of discrete states changing in time [6]. The time can be considered both discrete and continuous. The mentioned models can be characterized by relevant state vectors of system elements.

The reliability model of a modern complex technical system can contain up to several thousands of elements being in turn in different states (e.g. operating, failure, recovering) [7]. This results in large space of possible states in the corresponding Markov model [8]. It is convenient to describe the system reliability behavior by a graph, the nodes of which are attributed to the states of the system, and the edges to possible transitions from one state to another [9].

One of the classical approaches to the reliability calculations is a structurally logical analysis of technical systems. This approach for calculation of reliability parameters implies the use of structurally logical RBD [10] of technical systems, which graphically display the interconnection of elements and their influence on the system operability on the whole. Structurally logical diagram is a collection of previously allocated elements, connected between each other in series or in parallel. The criterion for the determination of the joint kind of the elements is the influence of their failure on the system operability. Under this approach the condition of the system operability can be defined with the help of the algebraic function of logics, i.e. in analyzing of the diagram topology it is first needed to determine such segments, which form the serial connection of elements and thus the joint method of the allocated segments between each other is being considered. In serial connection the elements are connected by the logical operation AND, in parallel connection – by the logical operation OR [11].

From the first sight it might seem that the execution of the above described operations is a quite simple task, which does not require special skills and knowledge and is of polynomial complexity. In reality it is not so, because even the construction of the operability condition for the technical system is not a simple task, especially in case of the availability of a collection of elements with diversified joints between them, while solving which manually there is a very high probability of a human error, particularly among designers with insufficient work experience. In its turn the construction and visualization of a state and transition graph is a sophisticated problem, because the number of possible states of the system depending on the number of elements, the number of element recoveries is growing exponentially which is in its turn reflected upon the complexity of the differential equation system.

DOI 10.15588/1607-3274-2020-4-16

By reference to the abovementioned the process of the reliable designing of complicated technical systems requires the automation of all its stages, starting from the construction of the RBD and ending with the visualization of the obtained results respectively and the task of automated calculation of the reliability indexes of the technical systems is currently central.

The next step, after the analysis of RDB and definition of a condition of the system operability of system, is a construction of a state matrix of a technical system. It contains the information on the operation of the system from the viewpoint of reliability [12]. Every matrix line is a vector, the components of which are the indicators of the state, in which each element remains, when the system itself is in state *n*. The element can be present only in two states: operating or restoring, it can also be found in a standby state, which can be caused by different reasons [13].

With reference to the availability of failures and recoveries the system randomly transits from one state to another and as such in the process of a long-term use it can be found in each of the possible states not once. In this case the process of its operation can be easily defined with the help of graphs, where the states of the system correspond to the graph nodes, and possible transitions from state to state (to the graph edges) [14, 15]. The use of Kolmogorov–Chaman differential equation system, see Eq. (1) allows to explain the reliability behavior of the system with time as a random process with the finite number of values and continuous replacement of the argument (time), i.e. the discrete-continuous stochastic process [16]. The solution of this equation system allows us to determine the reliability indices mentioned in the Sec. 1.

3 MATERIALS AND METHODS

Besides, for the solving of the problem for automation of reliability indexes calculation we have developed the method of automated formulation of operability condition of the technical systems based upon the RBD [17]. This method is based on three algorithms, each of them performs a specific functional role in the method operation and on the whole they allow to obtain the operability condition, based on the RBD for systems with undefined configuration of joints between the elements of the system (Fig. 1.)

As detailed above a state and transition graph is a convenient method for visual analysis of the system operating process, which allows to obtain the information on all possible states, the system can be found in (operating, standby, failure) as well as particular joints between the elements [18].

Thus, the algorithm of constructing a state and transition graph [18] was developed, which uses the operability condition of the system as an input parameter for the processing and determination of all possible states and joints between them based upon successive failures and recoveries of the system elements (Fig. 2).

We have also developed the algorithm [19] for the automation of the process for the workup of the Kolmogorov-Chapman differential equation system, which being based on the set of all system states and transitions allows to automatically generate and solve such equation systems without involvement of specialized software products (Matlab, Mathcad) for the analysis of structural RBD and automated determination of reliability indexes of complex technical systems (Fig. 3).

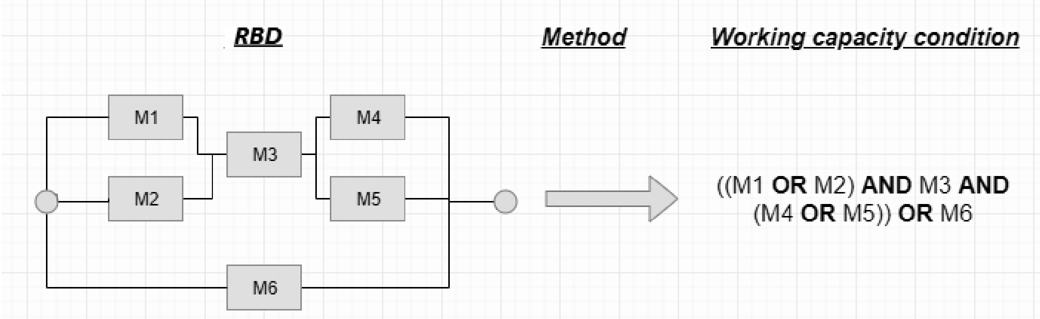


Figure 1 – The diagram of the operating method for the automated formulation of the operability condition of technical systems based on RBD

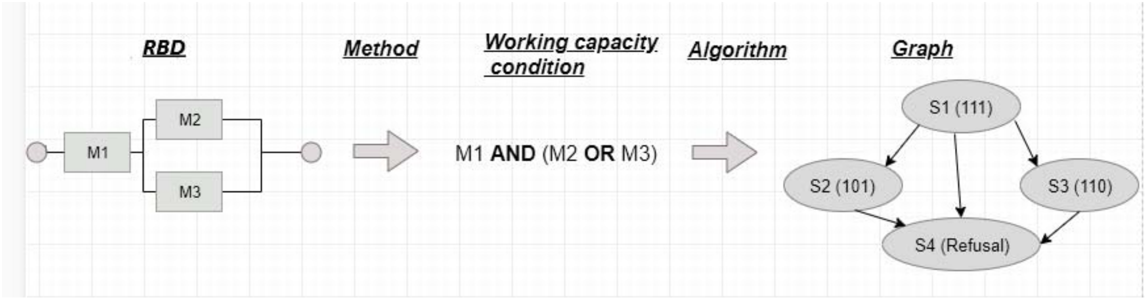


Figure 2 – Operating algorithm diagram [18]

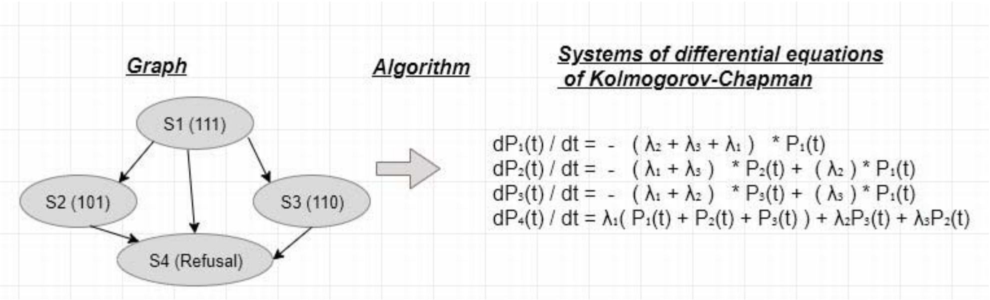


Figure 3 – Operating algorithm diagram

Thus it has become necessary to develop a number of algorithms and methods, that provide opportunities to automate the process for calculation of reliability indexes of complicated technical systems based upon RBD [17–19], as well as for the integration of already developed program modules [20] into the holistic software suite, which would allow to build RBD in an automated way for the description of operation of technical systems with different configuration and level of complexity and to formulate the operability conditions of such systems based upon them.

The software suite must provide the opportunity to construct a state and transition graph in an automated way as well as the differential equation system for the description of technical system operation. It also has to be able to make the calculations of reliability indexes of complex technical systems automatically as well as to visualize the obtained results.

As a result of analysis (RBD and Graph states and transitions) the formulas for the calculation of the minimum and maximum number of acceptable states for the system of *n* elements and *r* recoveries were derived:

– minimum number of acceptable states for the system of *n* elements and *n* recoveries – the number of states for serial connection of each of *n* elements and *r* recoveries, Eq. (2);

– maximum number of acceptable states for the system of *n* elements and *r* recoveries – the number of states for parallel connection of each of *n* elements and *r* recoveries, Eq. (3);

– the number of acceptable states for the system of *n* elements for the mixed type of connection between the system elements – this value is variable between the maximum and minimum permissible values depending on the configuration of joints between the elements in the system;

$$(2\cdot(r+1) - 1)^n + 1 \tag{2}$$

$$(2\cdot(r+1))^n \tag{3}$$

Let us consider the derivation of equations 2 and 3 in detail.

We have got a system of *n* elements (modules), which are arranged in parallel and each of the elements can be recovered *r* times (*r* is the same for each module).

The operation of this system can be represented with the help of a numeral, where *n* is the digit capacity, and *r* is an index – for each digit position order of module recovery (how many times it was already recovered).

So the operating state of the module corresponds to numeral 1 and the failure state to 0, but we should also take into account the index of the current recovery ($1_0\, 1_1\, 1_2\, 0_0\, 0_1\, 0_2 \ldots 1_r\, 0_r$), when increasing the number of recovery for the system by 1 ($r = r+1$), the set of elements for placement increases by 2, because the values $1_{r+1}\ 0_{r+1}$, therefore to determine the set of all possible states for parallel connection (consider – operating states – one of the elements in the system is operating, standby states – one or more elements are in the standby state and the others are in the state of failure or critical failure – all elements of the system are failed and cannot be recovered any more) we can use a specially adapted to this case combinatorial formula of arrangement with repetition:

$$A_n^m = n^m . \tag{4}$$

Therefore, Eq. (3) takes the following form:

$A_n^m = n^m = (2\cdot(r+1))^n$ – operating states, standby states, critical failure.

Eq. (3) also gives the opportunity to derive the equations for particular subsets of operating states and standby states, critical failure (this is always one state out of the overall set).

The state of critical failure for the system with *n* parallel arrangement of elements and with *r* recovery is as follows: $0_r\, 0_r\, 0_r \ldots 0_r$.

As mentioned above – the states of standby – when one or more elements are in the standby state, and others are in the state of failure. Then out of the set $\{0_0\, 0_1\, 0_2 \ldots 0_r\}$we can perform the repeated arrangement, which will include all standby states and one state of critical failure, therefore the formula for the determination of the number of standby states for the system with n parallel arrangement of elements and with *r* recovery is as follows (5):

$$(r+1)^n - 1 . \tag{5}$$

Then we need to subtract the number of operating states from the overall number of states to determine the number of operating states (6).

$$\begin{aligned}&(2\cdot(r+1))^n - ((r+1)^n - 1) - 1 = \\ &= 2^n \cdot (r+1)^n - (r+1)^n + 1 - 1 = (2^n - 1)\cdot(r+1)^n .\end{aligned} \tag{6}$$

© Yakovyna V. S., Seniv M. M., Symets I. I., Sambir N. B., 2020
DOI 10.15588/1607-3274-2020-4-16

Let us consider the example. We have got a system of 3 ($n = 3$) elements, which are arranged in parallel with the recovery value 1 ($r = 1$).

Then the set $\{1_0\, 1_1\, 0_0\, 0_1\}$ – of all possible elements for arrangement, and as far as in the system $n = 3$ – the number of elements that need to be arranged, that is why the number of states for this system is equal to:

$$A_n^m = n^m = (2 \cdot (r+1))^n = (2 \cdot (1+1))^3 = 64.$$

This value is the maximum value of all possible states for the system with $n = 3$ and $r = 1$, for systems with the same n and r values, but with serial or mixed element connection – the set of states will be less because this value is maximum for parallel connection between all elements of the system.

$(2 \cdot (r+1))^n - (r+1)^n - 2 = (2 \cdot 2)^3 - 2^n - 2 = 64 - 8 - 2 = 54$ – operating states out of 64 states

$(r+1)^n - 1 = (1+1)^3 - 1 = 8 - 1 = 7$ – standby states out of 64 states

For the system with serial arrangement of elements the formula for the determination of the set of states is calculated as follows.

We have a system with n elements (modules), which are arranged in serial and each of the elements can be recovered r times (r is the same for each module).

An important condition for a system with serial arrangement of elements is that we consider the operating states (all elements of the system are operating), standby states (one or more elements are in standby state and the others are operating), critical failure state (one element of the system is failed and cannot be recovered). It means that for the arrangement with repetition, which will also be used to derive the formula for the determination of the set of states for serial connection of elements in the system, the set of all possible elements for arrangement $\{1_0\, 1_1\, 1_2\, 0_0\, 0_1\, 0_2\, \ldots\, 1_r\, 0_r\}$ will be less by one, because we do not take into account $\{0_{\text{r index = r max}}\}$. $\{1_0\, 1_1\, 1_2\, 0_0\, 0_1\, 0_2\, \ldots\, 1_r\}$; out of this set we can define the arrangement for all operating and downtime states and the system will have 1 critical failure state, so the Eq. (2) will look as follows:

$$A_n^m = n^m = (2 \cdot (r+1) - 1)^n + 1\,.$$

Eq. (2) also gives the opportunity to derive the formulas for particular subsets of operating states and downtime states, critical failure (this is always one state out of the overall set).

Operating states for a system with n serial arrangement of elements and with r recoveries will consist of the arrangement with repetition out of the set of values $\{1_0\, 1_1\, 1_2\, \ldots\, 1_r\}$, therefore the formula for the determination of the number of operating states for a system with serial arrangement of elements will look as follows

$$(r+1)^n. \qquad (7)$$

The equation $(2 \cdot (r+1) - 1)^n$ allows to determine the total sum of operating and standby states for the system therefore the formula for the determination of the number of standby states with serial arrangement of elements will look like:

$$(2 \cdot (r+1) - 1)^n - (r+1)^n = (2r+1)^n - (r+1)^n. \qquad (8)$$

Let us consider an example. We have got a system of 3 ($n = 3$) elements, arranged in parallel with the recovery value 1 ($r = 1$).

Then the set {10 11 00 } – of all possible elements for arrangement (as far as 01 in arrangement for a serial system is not taken into account), and $n = 3$ is the number of elements that need to be arranged, therefore the number of states for this system is equal to: $A_n^m = n^m = (2 \cdot (r+1) - 1)^n + 1 = (2 \cdot (1+1) - 1)^3 + 1 = 28$.

This value is the minimum value of possible states for a system with $n = 3$ and $r = 1$, for a system with the same n and r values but with parallel or mixed connection of elements – the set of states will be greater.

$(r + 1)^n = 2^3 = 8$ – operating states out of 28 states

$(2r+1)^n - (r+1)^n = 3^3 - 2^3 = 19$ – standby states out of 28 states

The equations (2) and (3) as well as the corresponding equations (5)–(8) are suitable for fast estimation of the system for the maximum and minimum permissible number of states of the system with the determinate n and r values as far as we do not need any special methods, complicated calculations, etc. These equations were also tested in software via the software system, described in this article.

4 EXPERIMENTS

It has been decided that the developed software suite must have a modular architecture, where each of the modules allows to make calculations of particular features of a technical system (operability condition, state and transition graph, equation system, diagrams) independently of each other. A modular structure of the complex makes it flexible and allows to update or make modifications of necessary modules fast and without significant program changes (Fig. 4).

The suite contains one basic module, which is responsible for the interconnection and transfer of data between all other elements of the system. These modules can be divided into two groups:

1. Modules, responsible for the processing of particular features determination:
   - – Operability condition;
   - – State and transition graph;
   - – Differential equation system;
   - – Diagrams of numeric values of reliability indexes.
2. Modules, responsible for the visualization of information and configurations of graphic components.

For the implementation of functional capabilities of the software suite a flexible hierarchy of classes was de-

DOI 10.15588/1607-3274-2020-4-16

veloped. It allows making modifications easily by adding new functionality and modules.

The main class of the software suite is the class Reliability Schema. It allows us to make manipulations in the working space and in the space of system configurations. This is the basic class responsible for the entire operating of the system and its certain modules, included into it.

WorkabilityCondition – is the class, responsible for the construction of operability condition, it uses the algorithm of the reliability diagram bypass, the algorithms of successive and parallel joints of segments for the determination of an operability condition.

NodePath – is the class for the description of reliability diagram segments: the start and the end segment node, partial operability condition, segment modules, succession of a segment

SystemState – is the class for the determination of all possible states and transitions, in which the technical system under study might be present.

State – is the class for the description of the system state. It contains information on the type, order number, elements, which lead to failure.

SystemConfiguration – is a static class, that contains the information on the system configurations and features: system recovery; number of modules; deleted elements; construction of a state and transition graph; equation system composition; construction of diagrams and values of their definition.

We have processed the software with the help of Microsoft Visual Studio 2017 using the programming language C# and WPF technology for Windows platform (the software product works on the operating system Windows XP and higher). The initial stage of the software work is the creation and configuration of a structural scheme which describes the operation process of a technical system. The main window (Fig. 5) consists of a working space (Fig. 6) and configuration space.

The working space is intended for the RBD formulation. There are initialization nodes at the start and at the end of the window; the scheme must start and end at these points. The joint of elements in the scheme must be arranged from left to right.

The working space is represented by RBD for 12 elements with different joint types, which corresponds to the real technical system of a low complexity level. The configuration space is indented for the tracking and control of the RBD constructing process.

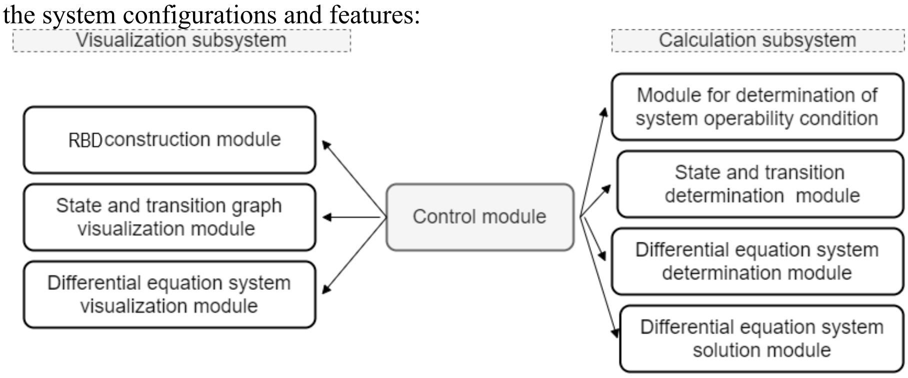
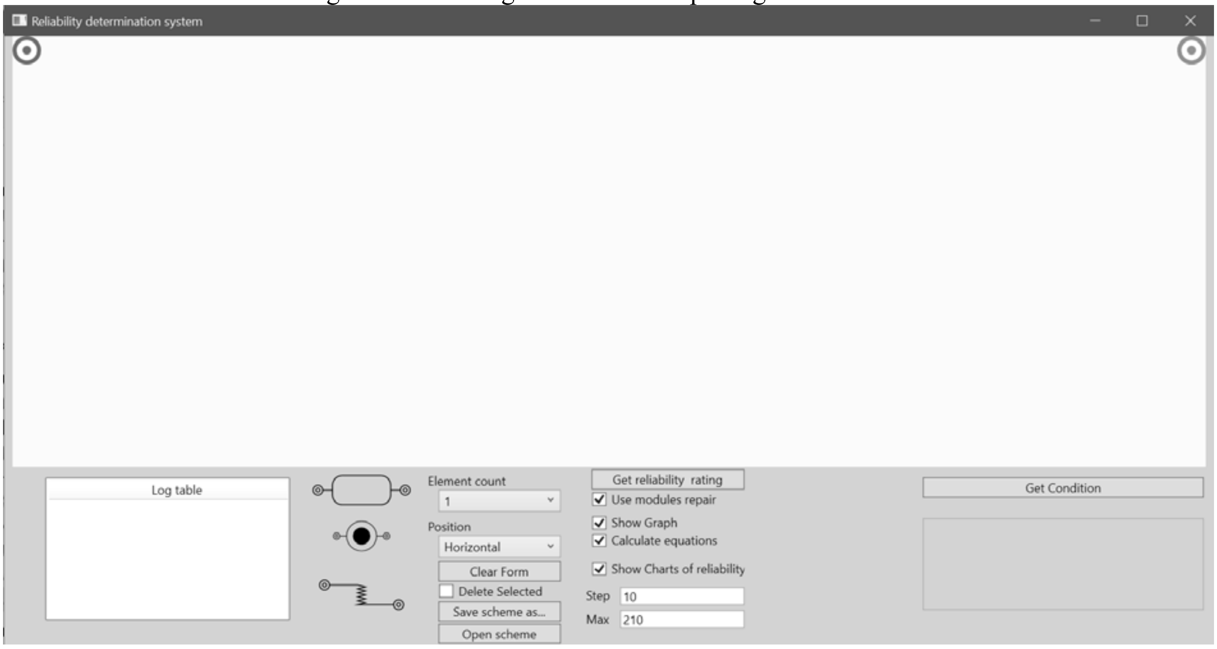


Figure 4 – The diagram of software package modules



Element count 1
Position Horizontal
☐ Delete Selected
☑ Use modules repair
☑ Show Graph
☑ Calculate equations
☑ Show Charts of reliability
Step 10
Max 210

Figure 5 – The main window of the software suite

DOI 10.15588/1607-3274-2020-4-16

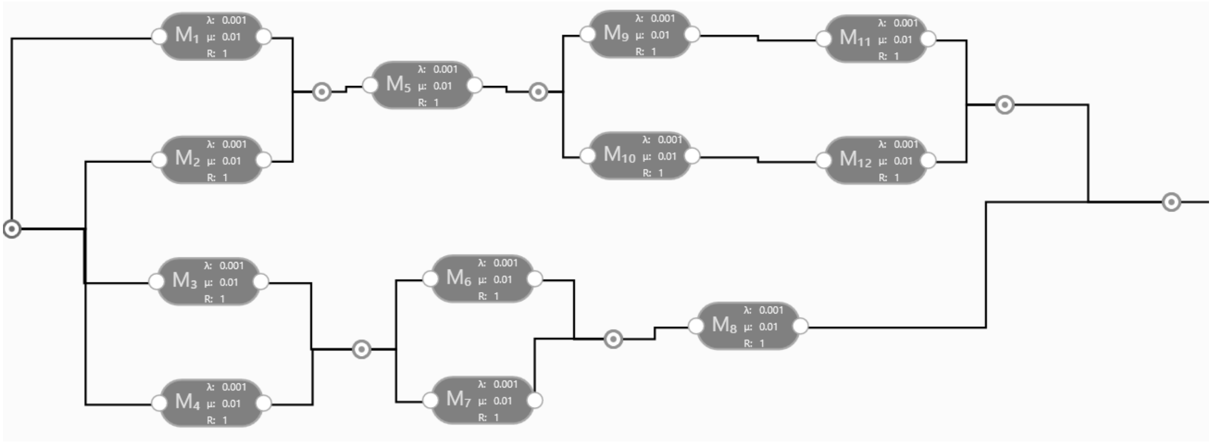
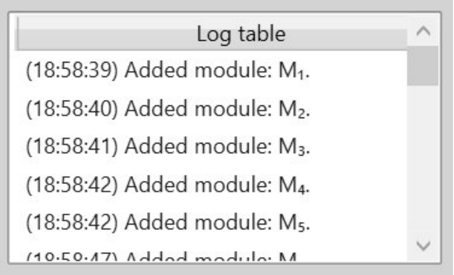


Figure 6 – Working space of the software suite

It includes the chronological chart of RBD graphic components creation (Fig. 7). This is a convenient method, which allows to track the process of adding components, the RBD consists of as well as it displays the time of their adding.



```
(18:58:39) Added module: M₁.
(18:58:40) Added module: M₂.
(18:58:41) Added module: M₃.
(18:58:42) Added module: M₄.
(18:58:42) Added module: M₅.
(18:58:47) Added module: M
```

Figure 7 – The chronological chart of the diagram graphic components creation

Besides, the configuration space allows to select the graphic element which is needed at a particular moment of time by clicking the mouse (1 – Module, 2 – Node, 3 – Line) (Fig. 8).

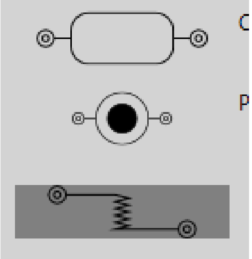
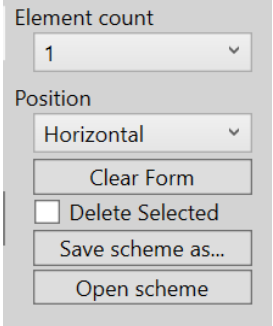


Figure 8 – Selection of a current graphic element for the RBD construction

As a matter of convenience, it is also possible to select the number of elements, which must be added simultaneously, the direction of their composition, clearing of the diagram completely or deleting of its particular elements (Fig. 9): element count – determines how many elements are built on the working frame by one click (refers to the module nodes); position – the direction of element composition: vertical or horizontal; clear form – complete clearing of the working window and all elements in it; delete selected – if a tick is put opposite this mark, then after clicking the right mouse button pointed at a particular element of the working window, it will be deleted; save schema as an open schema – allows to save and open a file with the saved RBD.



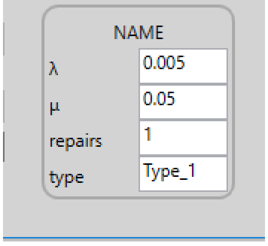
Element count: 1
Position: Horizontal

Figure 9 – Configurations of a diagram construction

The option of saving the RBD configuration is a convenient functionality while working with large systems, which allows to accelerate and improve the analysis process, because it makes it possible to get back to work over the analysis of a particular diagram fast at any convenient time.

The basic RBD component is a module. By default, each module has the following feature values (Fig. 10): probability of failure = 0.005; intensity of recovery =0.05; the number of recoveries = 1.

The user can change them by selecting a suitable module. To select the module, you need to click on it by the right mouse button (you should consider if the flag indicator next to Delete select is not selected).

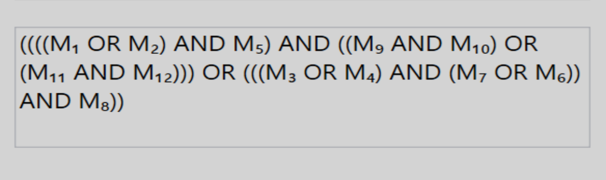


NAME
λ: 0.005
μ: 0.05
repairs: 1
type: Type_1

Figure 10 – Module features

DOI 10.15588/1607-3274-2020-4-16

After constructing the RBD for a particular system you can review the operability condition for this system (Fig. 11). To do this you need to click the button “Get Condition” in the configuration space and the result of the diagram analysis will appear in the window.



$((((M_1 \text{ OR } M_2) \text{ AND } M_5) \text{ AND } ((M_9 \text{ AND } M_{10}) \text{ OR } (M_{11} \text{ AND } M_{12}))) \text{ OR } (((M_3 \text{ OR } M_4) \text{ AND } (M_7 \text{ OR } M_6)) \text{ AND } M_8))$

Figure 11 – Operability condition for RBD

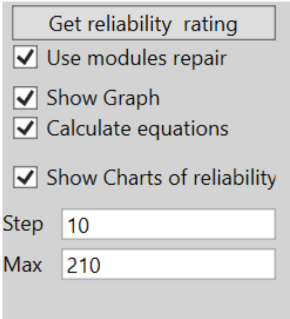
For further work with the saved RBD in a separate configuration window you can select the following options (Fig. 12):

– Use modules repair – use modules repair: if selected the modules repair is considered;

– Show graph – display state and transition graph;

– Calculate equations – build and solve the Kolmogorov-Chapman equation system;

– Show chart of reliability – construct the charts of reliability index number changes with the specification of tab spacing and timeline.



☑ Use modules repair

☑ Show Graph

☑ Calculate equations

☑ Show Charts of reliability

Step 10

Max 210

Figure 12 – Configuration window of reliability analysis parameters

After construction of RBD and selection of suitable calculation parameters you need to click “Get reliability rating” button. After a certain period the calculations will be made, the time of execution will be displayed in the logging window. Then the window with an open tab “Graph of states” will appear – this tab allows to display all existing states, the system can be found in as well as to view the general and detailed state and transition graph (Fig. 13).

This window is divided into three sections: state review section (1) – you can view the exact number of states, the system can be found in, and their description with the specified type; state details section (2) – allows to view each state in a more detailed interpretation with the description of states, which enter and leave it, as well as the states, which lead to a critical failure; state and transition graph review section (3) – this section allows you to see all possible states and transitions between them in a graphic representation, zoom in and scale down using the mouse wheel.

The software suite also provides the possibility to transit to a window, which allows to review the differential equation system and the chart of reliability index values of the system under study (Fig. 14). This window consists of two sections:

1. Section of the equation system view;
2. Section of the chart view:
   – probabilities of failure-free operation;
   – standby function;
   – probability of failure.

For the detailed review and analysis of solutions for the differential equation system you can transit to the tab “Runge-Kutta method”. Here you can review the information on the dependence of probability of system operation for the states at a particular moment of time (this chart is a detailed description of the equation system solutions). There is also an option of saving the chart data to a file or clipboard.

The program suite also provides an option to select the interface language from two possible variants: English and Ukrainian. To set the interface language you need to indicate which language you want to select in the App.config file (en-US or uk-Ua).

5 RESULTS

For verification of correct operation and validation of the developer software suite an integrated testing of all complex components and modules was carried out as well as the analysis of the developed algorithms operation was conducted [17–19] for a different set of input data with the estimation of the operation highspeed response. As far as the estimation of the highspeed response is one of the basic features of this kind of software which works with schemes, consisting of hundreds or thousands of elements.

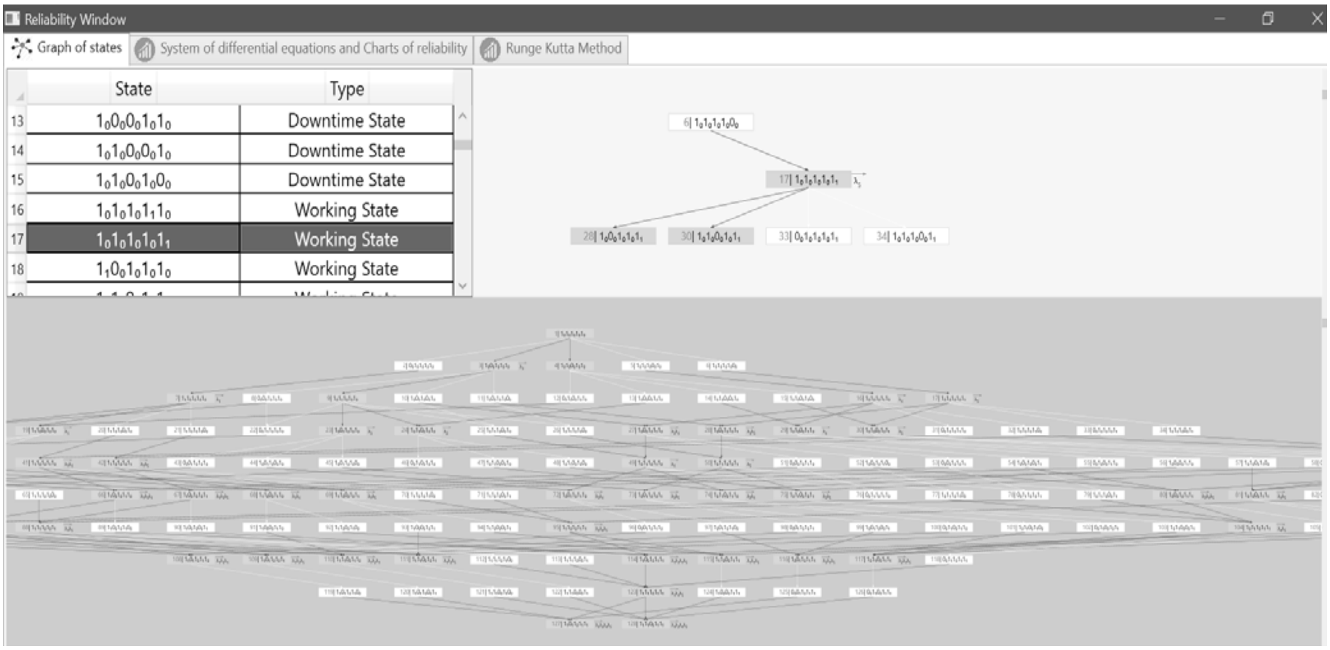
The main parameters that influence the highspeed response of the algorithms were determined as well as the analysis with different configuration and combination of these parameters was carried out. These parameters include the number of modules in the system, the number of joints between the modules in the system and the number of recoveries for one module. Testing of the system was carried out on a local machine with the following features: OS Windows 10, Intel Core i7, max 2.7 GHz, RAM 8 Gb DDR4 2133 MHz, 256 Gb SATA SSD, NVIDIA GeForce 940MX 2 Gb.

The analysis of the dependence of the highspeed response of method operation on the number of modules in the system for serial, parallel and mixed joints was carried out on three different system configurations:

– serial connection – all elements in the system are arranged in series one after another;

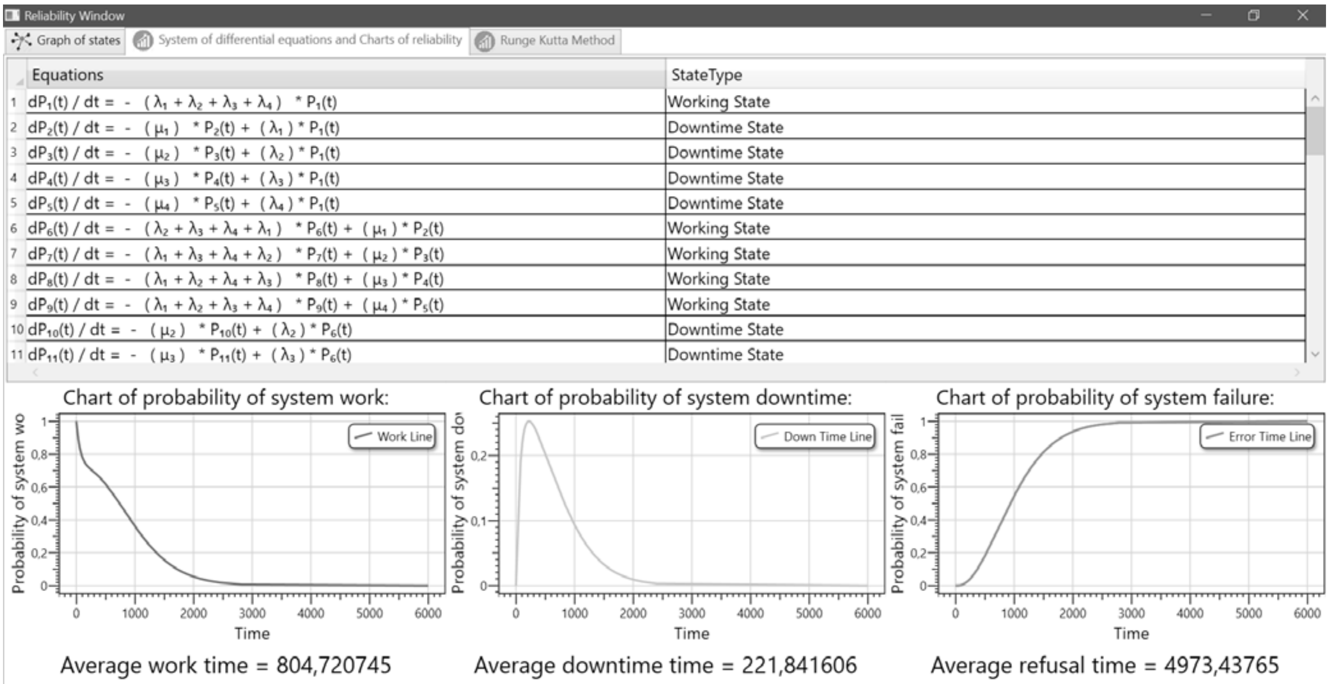
– parallel connection – all elements in the system are arranged in parallel;

– mixed connection – all elements in the system are arranged in different combinations of serial and parallel joints.



	State	Type
13	$1_0 0_0 0_0 1_0 1_0$	Downtime State
14	$1_0 1_0 0_0 0_0 1_0$	Downtime State
15	$1_0 1_0 0_0 1_0 0_0$	Downtime State
16	$1_0 1_0 1_0 1_1 1_0$	Working State
17	$1_0 1_0 1_0 1_0 1_1$	Working State
18	$1_1 0_0 1_0 1_0 1_0$	Working State

Figure 13 – "Graph of states" window



	Equations	StateType
1	$dP_1(t) / dt = - (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) * P_1(t)$	Working State
2	$dP_2(t) / dt = - (\mu_1) * P_2(t) + (\lambda_1) * P_1(t)$	Downtime State
3	$dP_3(t) / dt = - (\mu_2) * P_3(t) + (\lambda_2) * P_1(t)$	Downtime State
4	$dP_4(t) / dt = - (\mu_3) * P_4(t) + (\lambda_3) * P_1(t)$	Downtime State
5	$dP_5(t) / dt = - (\mu_4) * P_5(t) + (\lambda_4) * P_1(t)$	Downtime State
6	$dP_6(t) / dt = - (\lambda_2 + \lambda_3 + \lambda_4 + \lambda_1) * P_6(t) + (\mu_1) * P_2(t)$	Working State
7	$dP_7(t) / dt = - (\lambda_1 + \lambda_3 + \lambda_4 + \lambda_2) * P_7(t) + (\mu_2) * P_3(t)$	Working State
8	$dP_8(t) / dt = - (\lambda_1 + \lambda_2 + \lambda_4 + \lambda_3) * P_8(t) + (\mu_3) * P_4(t)$	Working State
9	$dP_9(t) / dt = - (\lambda_1 + \lambda_2 + \lambda_3 + \lambda_4) * P_9(t) + (\mu_4) * P_5(t)$	Working State
10	$dP_{10}(t) / dt = - (\mu_2) * P_{10}(t) + (\lambda_2) * P_6(t)$	Downtime State
11	$dP_{11}(t) / dt = - (\mu_3) * P_{11}(t) + (\lambda_3) * P_6(t)$	Downtime State

Average work time = 804,720745

Average downtime time = 221,841606

Average refusal time = 4973,43765

Figure 14 – The view window of the differential equation system and the chart of reliability index values

For program implementation of the method for automated formulation of operability condition of technical systems based upon the RBD the analysis with different number of modules in the system and different configuration of joints between the modules (parallel and serial) was carried out. The results of the measurement of the operating time are represented in the Table 1 and Fig. 15. The number of recoveries for this method is not a determining factor that influences the operation highspeed response, as it is not taken into account at this step.

In its turn the operation analysis of the program implementation for the algorithm of construction a state and transition graph was carried out for the schemes of different configurations, but this kind of analysis implies that the number of system module recoveries should be considered, because this factor has a strong effect on the number of states and transitions between them. The Table 2 and Fig. 16 displays the effect of the number of recoveries for the system of 5 modules with serial and parallel connection on the number of states in a graph (operating state, standby and critical failure are considered).

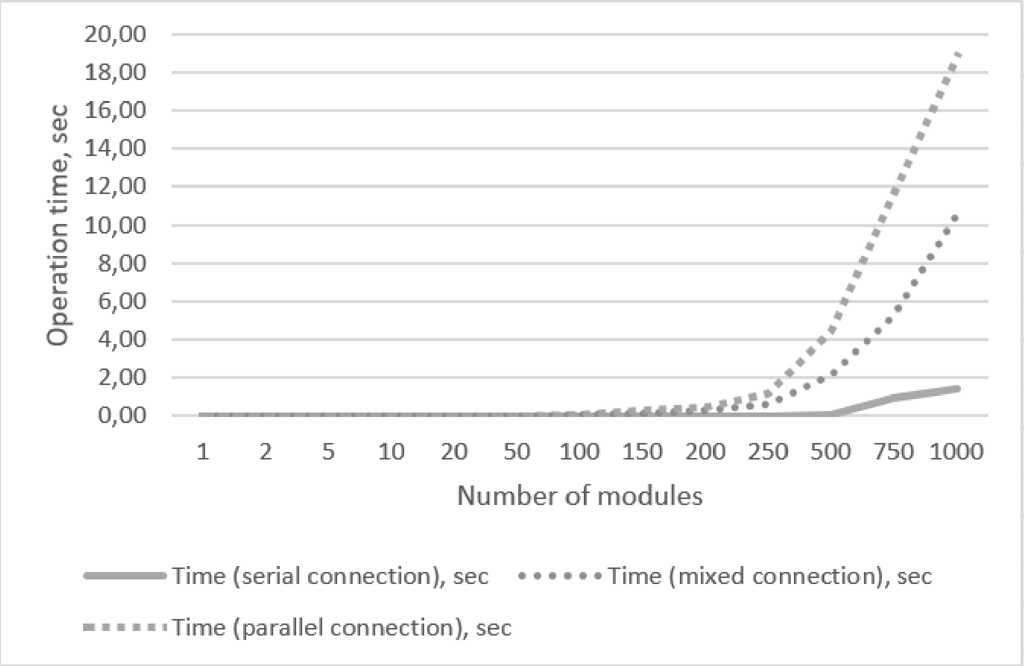


Figure 15 – Diagram of dependence of the operating time of method operation on the number of modules in the system for serial, parallel and mixed connections

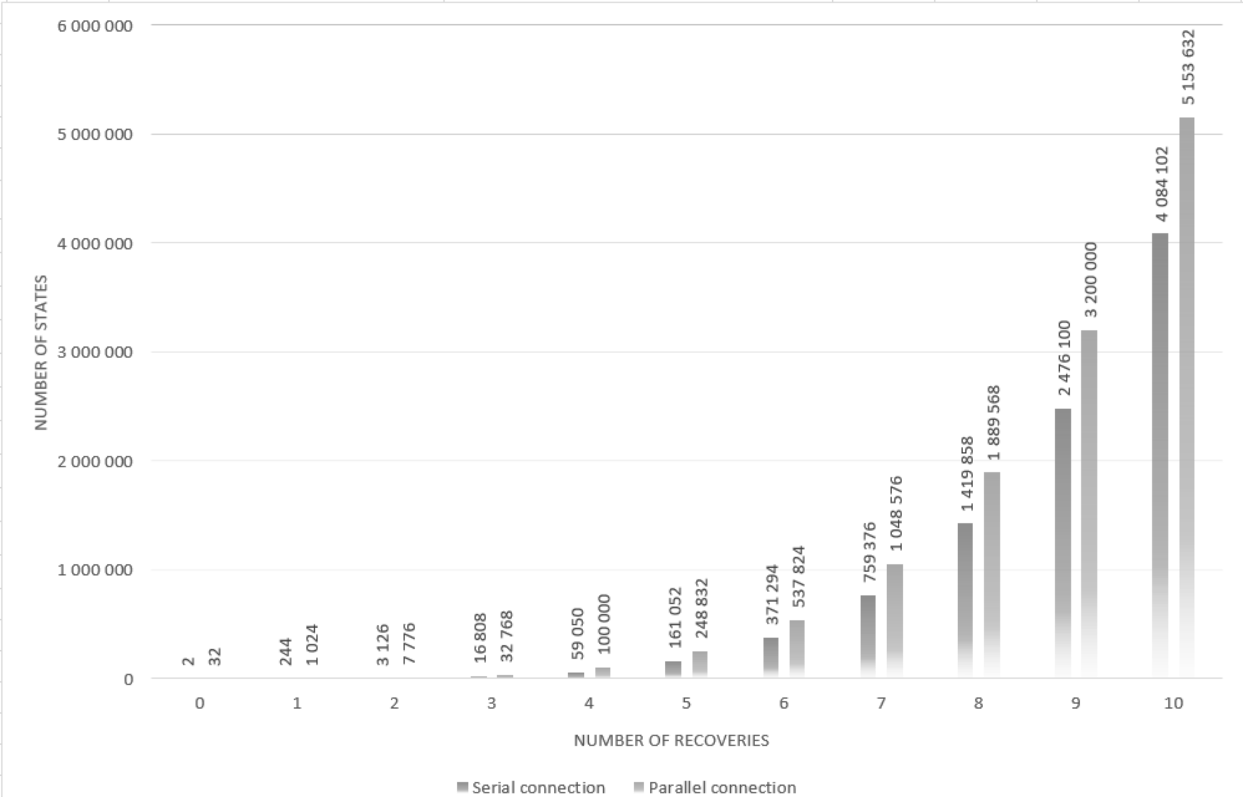


Figure 16 – Column chart on the dependence of the number of states of each element in a graph for the system of 5 modules on the number of recoveries for parallel and serial connection between the elements of the system

DOI 10.15588/1607-3274-2020-4-16

Table 1 – Operation time analysis of the method for automated formulation of technical systems working condition implementation.

№	Number of modules	Time (serial connection), sec	Time (mixed connection), secc	Time (parallel connection), sec
1	1	0.0000308	0.0000321	0.0000330
2	2	0.0000474	0.0000484	0.0000488
3	5	0.0000791	0.0000821	0.0000922
4	10	0.0001703	0.0002093	0.0002418
5	20	0.0002004	0.0004931	0.0007001
6	50	0.0003784	0.0058938	0.0037932
7	100	0.0004165	0.0486683	0.0571824
8	150	0.0004484	0.1509184	0.2522243
9	200	0.0005171	0.3154700	0.4550050
10	250	0.0060551	0.6377826	1.1881978
11	500	0.0106783	2.1342265	4.4533016
12	750	0.9105432	5.2297644	11.6021298
13	1000	1.4004321	10.5671256	18.8623663

Table 2 – The number of states in a graph upon the recoveries number of each element for the system of 5 modules

№	Number of recoveries	Number of states in serial connection	Number of states in parallel connection
1	0	2	32
2	1	244	1 024
3	2	3 126	7 776
4	3	16 808	32 768
5	4	59 050	100 000
6	5	161 052	248 832
7	6	371 294	537 824
8	7	759 376	1 048 576
9	8	1 419 858	1 889 568
10	9	2 476 100	3 200 000
11	10	4 084 102	5 153 632

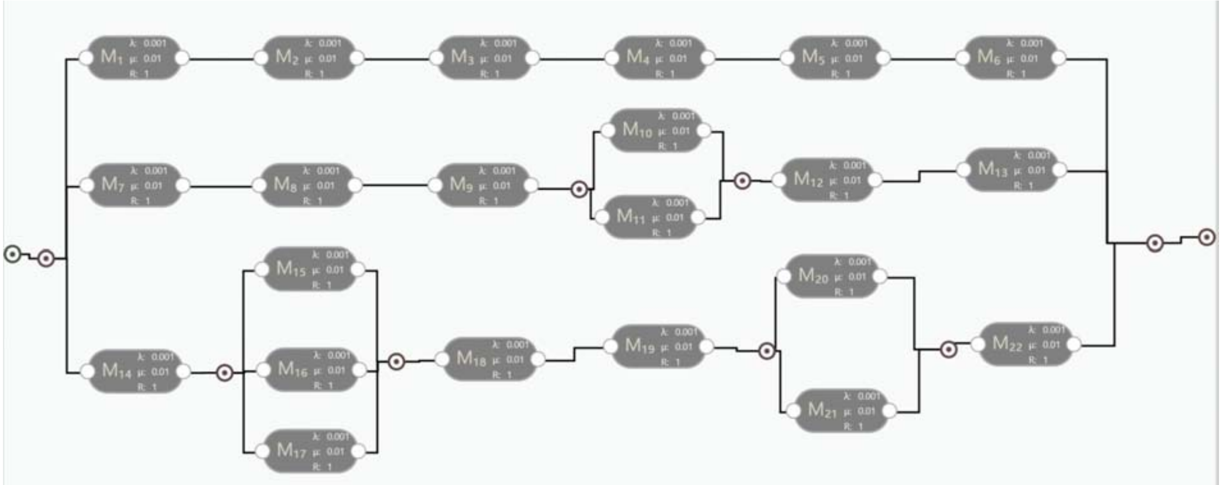


Figure 17 – The reliability block diagram for the unrecoverable system of 22 elements

The analysis of the operation time of software implementation of the algorithm for the construction of a state and transition graph was carried out for a system with parallel connection of modules without the recovery of the module states with the increase of the number of modules in the system (Table 3).

Table 3 – The operation time of software implementation of the algorithm for construction of state and transition graph

№	Number of modules	Number of states	Algorithm operating time, sec
1	1	2	0.0451121
2	2	4	0.0119204
3	5	32	0.3929208
4	10	1024	3.3207597
5	15	32768	8.6198466
6	20	1048576	58.1923007
7	25	33554432	> 6 min

In its turn the operation of the algorithm for processing of the Kolmogorov–Chapman differential equation system consists of two parts:

– determination of equations – this process consists of the determination of equations based on the Eq. (1) – for each state of the system.

– solving the equation system using the Runge-Kutta method.

The operation time of software implementation of the algorithm for processing of the Kolmogorov-Chapman differential equation system is presented in the Table 4 for systems with the number of states varying from 1,000 up to 1,000,000. It is easy to notice that the developed software suite can solve the differential equation system for 106 states in a quite short time (near 35 sec).

© Yakovyna V. S., Seniv M. M., Symets I. I., Sambir N. B., 2020
DOI 10.15588/1607-3274-2020-4-16

Table 4 – The operation time of the equation workup process

№	Number of states	Processing time, sec.
1	1,000	0.023
2	10,000	0.586
3	50,000	1.127
4	100,000	2.012
5	250,000	4.827
6	500,000	10.142
7	750,000	18.721
8	1,000,000	35.299

6 DISCUSSION

As seen from the Table 1 of the dependence of the method's operation time on the number of modules in the system for serial, parallel and mixed connections the developed software suite provides the opportunity to formulate the working condition for the system of 1000 elements (modules) with a mixed type of connection (Fig. 15) during the short time, which corresponds to a rather complex technical system with the most widely used type of connection between the system elements (mixed).

The obtained data (see Table 2 and Table 3) allow us to conclude that the visualization of a state and transition graph for technical systems with a large number of modules is a sophisticated task (depending on the configuration of the system; for example the number of possible states for a system of 30 elements, arranged in parallel is about $1.07\ 10^9$). The time of operation of the software suite is growing, and thus there is a need of decomposition of such technical systems into smaller subsystems.

For example, Fig. 17 displays the RBD for an unrecoverable system of 22 elements. Time for determination of working condition for this technical system is about 0.0004 sec. Time for execution of the algorithm for construction of a state and transition graph (the number of determined states is 52 694) is 35.168 sec. Time for processing of the Kolmogorov-Chapman differential equation system for this system is 1.544 sec.

Total time for execution of a particular task for the reliability assessment of technical system consisting of 22 elements using RBD is 36.712 sec. The most time was given to the construction of a state and transition graph and this should be improved to increase the operation time of the developed software suite.

CONCLUSIONS

This paper presents algorithms and methods, that provide opportunities to automate the process for calculation of reliability indexes of complicated technical systems based upon RBD , as well as formulas for estimating the total and working number of system states based on input indicators about the system n and r.

According to the mathematical apparatus, a software was developed for the calculation of reliability indexes of complicated technical systems, which provides opportunities to automate the process of their reliability analysis, reduces the influence of a human factor and thus reduces the probability of making errors on early stages of reliability indexes calculation.

The developed software package has a modular architecture, where each of the modules makes it possible to carry out the calculation of particular reliability features of a technical system (operability condition, a state and transition graph, equation system, graphs of results visualization) independently of each other. Modular structure of the software suite makes it flexible and gives an opportunity to add and make modifications of modules fast and without significant program changes.

This software suite allows us to easily construct RBD for a technical system, to determine the operability condition automatically (execution time ≈ 10 sec for 1,000 elements with mixed type of connection), to form a state/transition matrix (up to 10^9 states), to form the differential equation system automatically and solve it (execution time ≈ 35 sec for 10^9 states) and, thus to obtain the numerical values of reliability indexes (probabilities of failure-free operation and probabilities of failure and standby).

Besides, the developed software suite is cross-platform, it has a convenient and ergonomic interface, Ukrainian and English localization, and other additional functions for the convenience of a design engineer.

The scientific novelty of the results presented in the paper lies in the further development of Markov reliability modelling and in firstly obtained model for the number of operational and failure states for the system of n elements and r recoveries using RBD approach.

The practical significance of the described study lies in the developed algorithms and software suite which makes it possible the automated reliability assessment of complex technical systems with up to 1,000 elements forming up to 1,000,000 states during the time less than one minute.

The future research will be focused on improvement the calculational speed of the developed software suite by developing appropriate methods and algorithms.

REFERENCES

1. Pham H. System software reliability. London, Springer, 2006, 440 p. DOI : 10.1007/1-84628-295-0
2. Fowler K. Dependability [reliability], *IEEE Instrumentation & Measurement Magazine*, 2005, Volume 8, Issue 4, pp. 55–58. doi: 10.1109/MIM.2005.1518623
3. Mulyak A., Yakovyna V., Volochiy B. Influence of software reliability models on reliability measures of software and hardware systems, *Eastern-European Journal of Enterprise Technologies*, 2015, Volume 4, Issue 9, pp. 53–57. DOI : 10.15587/1729-4061.2015.47336
4. Catelani M., Ciani L., Luongo V. A simplified procedure for the analysis of Safety Instrumented Systems in the process industry application, *Microelectronics Reliability*, 2011. Volume 51, Issues 9–11, pp. 1503–1507. DOI: 10.1016/j.microrel.2011.07.044
5. Haxthausen A. E. A Domain-Specific Framework for Automated Construction and Verification of Railway Control Systems, *Computer Safety, Reliability, and Security : International conference SAFECOMP 2009, Hamburg, 15–18 September 2009 : proceedings*. Berlin, Springer, 2009, pp. 1–3. DOI : 10.1007/978-3-642-04468-7_1
6. Lyu M. R. Software Reliability Engineering: A Roadmap, *Future of Software Engineering (FOSE'07), Minneapolis, 23–25*

© Yakovyna V. S., Seniv M. M., Symets I. I., Sambir N. B., 2020
DOI 10.15588/1607-3274-2020-4-16

May 200 , proceedings. Minneapolis, IEEE, 2007, pp. 153–170. DOI : 10.1109/FOSE.2007.24

7. Catelani M., Ciani L., Venzi M. Improved RBD analysis for reliability assessment in industrial application, *International Instrumentation and Measurement Technology (I2MTC) : 2014 IEEE Conference, Montevideo, 12–15 May 2014 : proceedings*. Montevideo, IEEE, 2014, pp. 670–674. DOI : 10.1109/I2MTC.2014.6860827
8. Pérez-Rosés H. Sixty Years of Network Reliability, *Mathematics in Computer Science,* 2018, Volume 12, Issue 3, pp. 275–293. DOI : 10.1007/s11786-018-0345-5
9. Brown J. I., Cox D., Ehrenborg R.The average reliability of a graph, *Discrete Applied Mathematics*, 2014, Volume 177, pp. 19–33. DOI : 10.1016/j.dam.2014.05.048
10. Sahinoglu M., Ramamoorthy C. V. RBD tools using compression, decompression, hybrid techniques to code, decode, and compute reliability in simple and complex embedded systems, *IEEE Transactions on Instrumentation and Measurement,* 2005, Volume 54, Issue 5, pp. 1789–1799. DOI: 10.1109/TIM.2005.855103
11. Bennetts R. G. Analysis of Reliability Block Diagrams by Boolean Techniques, *IEEE transactions on reliability*, 1982. Volume R-31, Issue 2, pp. 159–166. DOI : 10.1109/TR.1982.5221283
12. Catelani M., Ciani L., Venzi M. Reliability assessment for complex systems: A new approach based on RBD models, *Systems Engineering (ISSE), 2015 IEEE International Symposium, Rome, 28–30 September 2015, proceedings*. Rome, IEEE, 2015, pp. 286–290. DOI: 10.1109/SysEng.2015.7302771
13. da Silva Neto A. V., Vismari L. F., Gimenes R. A. V. et al. A Practical Analytical Approach to Increase Confidence in PLD-Based Systems Safety Analysis, *IEEE Systems Journal,* 2018, Volume 12, Issue 4, pp. 3473–3484. DOI : 10.1109/JSYST.2017.2726178
14. Tannous O., Xing L., Rui P. et al. Redundancy allocation for series-parallel warm-standby systems, *Industrial Engineering and Engineering Management, 2011 IEEE International Conference, Singapore, 6–9 December 2011, proceedings*. Singapore, IEEE, 2011, pp. 1261–1265. DOI: 10.1109/IEEM.2011.6118118
15. Teslyuk T., Teslyuk V., Denysyuk P. et al. Synthesis of Neurocontroller for Intellectualization Tasks of Process Control Systems, *The Experience of Designing and Application of CAD Systems (CADSM), 2019 IEEE 15th International Conference, Polyana, 26 February – 2 March 2019 : proceedings*. Lviv, IEEE, 2019, pp. 1–4. doi : 10.1109/CADSM.2019.8779295
16. Catelani M., Ciani L., Luongo V. A new proposal for the analysis of safety instrumented systems. *Instrumentation and Measurement Technology, 2012 IEEE International Conference, Graz, 13–16 May 2012, proceedings.* Graz, IEEE, 2012, pp. 1612–1616. DOI : 10.1109/I2MTC.2012.6229556
17. Bobalo Yu. Ya., Seniv M. M., Yakovyna V. S. et al. Method of Reliability Block Diagram Visualization and Automated Construction of Technical System Operability Condition, *Advances in Intelligent Systems and Computing III*, 2019, Volume 871, pp. 599–610. DOI : 10.1007/978-3-030-01069-0_43
18. Bobalo Yu., Yakovyna V., Seniv M. et al. Technique of automated construction of states and transitions graph for the analysis of technical systems reliability, *Computer Science and Information Technologies CSIT-2018, XIII International Scientific and Technical Conference*, Lviv, 11–14 September 2018 : proceedings. Lviv, IEEE, 2018, pp. 314–317. doi : 10.1109/STC-CSIT.2018.8526698
19. Bobalo Yu., Yakovyna V., Seniv M. et al. Techniques of automated processing of Kolmogorov – Chapman differential equation system for reliability analysis of technical systems, *The Experience of Designing and Application of CAD Systems (CADSM), 2019 IEEE 15th International Conference*, Polyana, 26 February – 2 March 2019, proceedings. Lviv : IEEE, 2019. – P. 267–272. DOI : 10.1109/CADSM.2019.8779271
20. Seniv M., Yakovyna V., Symets I. Software for visualization of reliability block diagram and automated formulation of operability conditions of technical systems, *Perspective Technologies and Methods in MEMS design MEMSTECH'2018 : XIV International Conference*, Lviv-Polyana, 18–22 April 2018 : proceedings. Lviv, IEEE, 2018. pp. 191–195. doi : 10.1109/MEMSTECH.2018.8365731

Received 10.10.2020.
Accepted 13.11.2020.

УДК 004.93

АЛГОРИТМИ ТА ПРОГРАМНИЙ ЗАСІБ ДЛЯ ОЦІНЮВАННЯ НАДІЙНОСТІ СКЛАДНИХ ТЕХНІЧНИХ СИСТЕМ

Яковина В. С.[1, 2] – [1] д-р. техн. наук, професор, професор кафедри систем штучного інтелекту, Національний університет «Львівська політехніка», Львів, Україна.

[2] Факультет математики та комп’ютерних наук, Вармінсько-Мазурський Університет в Ольштині, Польща.

Сенів М. М. – канд. техн. наук, доцент, доцент кафедри програмного забезпечення, Національний університет «Львівська політехніка», Львів, Україна.

Симець І. І. – асистент кафедри програмного забезпечення, Національний університет «Львівська політехніка», Львів, Україна.

Самбір Н. Б. – асистент кафедри програмного забезпечення, Національний університет «Львівська політехніка», Львів, Україна.

АНОТАЦІЯ

Актуальність. Однією з найважливіших властивостей технічних систем є їх надійність, тобто здатність системи виконувати задані функції, зберігаючи в часі значення експлуатаційних показників в заданих межах. Вартість відмови для сучасних складних технічних систем може бути дуже високою, що може призвести до подій різного ступеня тяжкості, починаючи від економічних збитків і закінчуючи шкодою для здоров’ю і життя людей. Отже, вимоги до їх надійності постійно зростають. Процес оцінки надійності складних технічних систем можна спростити поєднанням аналітичних методів дослідження з обчислювальними можливостями сучасних комп’ютерів. Найпоширеніші аналітичні методи базуються на теорії марковських процесів, яка в свою чергу дають можливість визначити часові залежності ймовірностей перебування системи у визначених станах (працездатності, відновлення, відмови), а отже, значення і часові залежності необхідних показників надійності. Ці методи можуть бути використані для аналізу надійності технічних систем різного типу: невідновлюваних та відновлюваних; резервованих та нерезервованих з різним видом резервування чи типом обслуговування та ін. Однак застосування цих методів для складних технічних систем, що містять велику кількість елементів вимагає виконання великої кількості розрахунків, що унеможливлює виконання цих завдань вручну. Отже, проблема автоматизації моделювання надійнос-

DOI 10.15588/1607-3274-2020-4-16

ті складної технічної системи з використанням сучасних обчислювальних систем є дуже актуальною темою дослідження. Для вирішення цієї проблеми можна використовувати конкретні алгоритмічні та програмні прийоми, описані в цій роботі.

Мета. Метою даної статті є розробка алгоритмів автоматизованої обробки структурних схем надійності (ССН) та оцінки показників надійності складних технічних систем поряд із набором програм для автоматизованої оцінки надійності.

Метод. Для проведення аналізу надійності використовується підхід ССН, який дозволяє представляти та візуалізувати кожен елемент системи у вигляді прямокутника, з'єднаного лініями паралельно або послідовно з іншими елементами системи. Для отримання значень показників надійності запропоновано математичну модель поведінки надійності технічної системи з використанням випадкового марковського процесу. Далі розглянуто алгоритм обробки ССН та автоматичного визначення умови працездатності технічної системи. Для розрахунку мінімальної та максимальної кількості робочих станів та станів відмов для системи з n елементів та r відновлення в роботі вводиться математична модель, заснована на комбінаторному підході. Для розробки програмного комплексу був використаний об'єктно-орієнтований підхід.

Результати. Набір алгоритмів та програмного забезпечення дозволяє нам легко побудувати ССН для технічної системи, автоматично визначити стан працездатності з часом виконання близько 10 сек. для 1000 елементів із змішаним типом з'єднання, автоматично сформувати матрицю станів та переходів разом із відповідною системою диференціальних рівнянь та вирішити її із загальним часом виконання близько 35 сек. для 109 станів і, таким чином, отримати числові значення показників надійності для досліджуваної технічної системи. Дослідження оцінки надійності для системи, що складається з 22 елементів із використанням ССН, показує, що загальний час виконання програмної реалізації становить 36,712 сек. Під час виконання цього тестового випадку найбільше часу (35,168 сек.) було витрачено на роботу алгоритму побудови графа станів та переходів, що складається з 52 694 станів.

Висновки. Представлені алгоритми та методи автоматизованої оцінки показників надійності складних технічних систем на основі підходу ССН, а також модель для визначення кількості станів системи(також включає визначення працездатних станів і станів відмови). Модульна структура розробленого набору програм робить його гнучким та дає можливість додавати та вносити модифікації модулів швидко та без значних змін програми.

КЛЮЧОВІ СЛОВА: програмне забезпечення, надійність, структурна схема надійності, граф станів та переходів.

УДК 004.93

АЛГОРИТМЫ И ПРОГРАММНОЕ СРЕДСТВО ДЛЯ ОЦЕНИВАНИЯ НАДЕЖНОСТИ СЛОЖНЫХ ТЕХНИЧЕСКИХ СИСТЕМ

Яковина В. С.[1, 2] – [1] д-р техн. наук, профессор, профессор кафедры систем искусственного интеллекта, Национальный университет «Львовская политехника», Львов, Украина.

[2] Факультет математики и компьютерных наук, Варминско-Мазурский Университет в Ольштыне, Польша.

Сенив М. М. – канд. техн. наук, доцент, доцент кафедры программного обеспечения, Национальный университет «Львовская политехника», Львов, Украина.

Симец И. И. – ассистент кафедры программного обеспечения, Национальный университет «Львовская политехника», Львов, Украина.

Самбир Н. Б. – ассистент кафедры программного обеспечения, Национальный университет «Львовская политехника», Львов, Украина.

АННОТАЦИЯ

Актуальность. Одним из важнейших свойств технических систем является их надежность, то есть способность системы выполнять заданные функции, сохраняя во времени значения эксплуатационных показателей в заданных пределах. Стоимость отказа для современных сложных технических систем может быть очень высокой, что может привести к событиям различной степени тяжести, начиная от экономических убытков и заканчивая ущербом для здоровья и жизни людей. Итак, требования к их надежности постоянно растут. Процесс оценки надежности сложных технических систем можно упростить сочетанием аналитических методов исследования с вычислительными возможностями современных компьютеров. Самые распространенные аналитические методы базируются на теории марковских процессов, которая в свою очередь дает возможность определить временные зависимости вероятностей нахождения системы в определенных состояниях (работоспособности, восстановления, отказа), а следовательно, значение и временные зависимости необходимых показателей надежности. Эти методы могут быть успешно использованы для анализа надежности различных технических систем: как невосстанавливаемых, так и восстанавливаемых; без резервирования и с резервированием различных типов, приоритетов обслуживания и т. д. Однако применение этих методов для сложных технических систем, содержащих большое количество элементов требует выполнения большого количества расчетов, делает невозможным выполнение этих задач вручную. Итак, проблема автоматизации моделирования надежности сложной технической системы с использованием современных вычислительных систем является очень актуальной темой исследования. Для решения этой проблемы можно использовать конкретные алгоритмические и программные приемы, описанные в этой работе.

Цель. Целью данной статьи является разработка алгоритмов автоматизированной обработки структурных схем надежности (ССН) и оценки показателей надежности сложных технических систем наряду с набором программ для автоматизированной оценки надежности.

Метод. Для проведения анализа надежности используется подход ССН, который позволяет представлять и визуализировать каждый элемент системы в виде прямоугольника, соединенного линиями параллельно или последовательно с другими элементами системы. Для получения значений показателей надежности предложена математическая модель поведения надежности технической системы с использованием случайного марковского процесса. Далее рассмотрен алгоритм обработки ССН и автоматического определения условия работоспособности технической системы. Для расчета минимального и максимального количества рабочих состояний и состояний отказов для системы с n элементов и r восстановлений в работе вводится математическая модель, основанная на комбинаторном подходе. Для разработки программного комплекса был использован объектно-ориентированный подход.

DOI 10.15588/1607-3274-2020-4-16

Результаты. Набор алгоритмов и программного обеспечения позволяет нам легко построить ССН для технической системы, автоматически определить состояние работоспособности со временем выполнения около 10 сек на 1000 элементов со смешанным типом соединения, автоматически сформировать матрицу состояний и переходов вместе с соответствующей системой дифференциальных уравнений и решить ее с общим временем выполнения около 35 сек для 109 состояний и, таким образом, получить числовые значения показателей надежности для исследуемой технической системы. Исследование оценки надежности для системы, состоящей из 22 элементов с использованием ССН, показывает, что общее время выполнения программной реализации составляет 36,712 сек. Во время выполнения этого тестового случая больше времени (35,168 сек) было потрачено на работу алгоритма построения графа состояний и переходов, состоящего из 52 694 состояний.

Выводы. Представленные алгоритмы и методы автоматизированной оценки показателей надежности сложных технических систем на основе подхода ССН, а также модель для определения количества состояний системы (также включает определение трудоспособных состояний и состояний отказа). Модульная структура разработанного набора программ делает его гибким и позволяет добавлять и вносить модификации модулей быстро и без значительных изменений программы.

КЛЮЧЕВЫЕ СЛОВА: программное обеспечение, надежность, структурная схема надежности, граф состояний и переходов.

ЛІТЕРАТУРА / ЛИТЕРАТУРА

1. Pham H. System software reliability / H. Pham. – London : Springer, 2006. – 440 p. doi : 10.1007/1-84628-295-0
2. Fowler K. Dependability [reliability] / K. Fowler // IEEE Instrumentation & Measurement Magazine. – 2005. – Volume 8, Issue 4. – P. 55–58. DOI: 10.1109/MIM.2005.1518623
3. Mulyak A. Influence of software reliability models on reliability measures of software and hardware systems / A. Mulyak, V. Yakovyna, B. Volochiy // Eastern-European Journal of Enterprise Technologies. – 2015. – Volume 4, Issue 9. – P. 53–57. DOI : 10.15587/1729-4061.2015.47336
4. Catelani M. A simplified procedure for the analysis of Safety Instrumented Systems in the process industry application / M. Catelani, L. Ciani, V. Luongo // Microelectronics Reliability. – 2011. – Volume 51, Issues 9–11. – P. 1503–1507. DOI : 10.1016/j.microrel.2011.07.044
5. Haxthausen A. E. A Domain-Specific Framework for Automated Construction and Verification of Railway Control Systems / A. E. Haxthausen // Computer Safety, Reliability, and Security : International conference SAFECOMP 2009, Hamburg, 15–18 September 2009 : proceedings. – Berlin : Springer, 2009. – P. 1–3. DOI : 10.1007/978-3-642-04468-7_1
6. Lyu M. R. Software Reliability Engineering: A Roadmap / M. R. Lyu // Future of Software Engineering (FOSE'07), Minneapolis, 23–25 May 2007 : proceedings. – Minneapolis : IEEE, 2007. – P. 153–170. DOI : 10.1109/FOSE.2007.24
7. Catelani M. Improved RBD analysis for reliability assessment in industrial application / M. Catelani, L. Ciani, M. Venzi // International Instrumentation and Measurement Technology (I2MTC) : 2014 IEEE Conference, Montevideo, 12–15 May 2014 : proceedings. – Montevideo : IEEE, 2014. – P. 670–674. DOI : 10.1109/I2MTC.2014.6860827
8. Pérez-Rosés H. Sixty Years of Network Reliability / H. Pérez-Rosés // Mathematics in Computer Science. – 2018. – Volume 12, Issue 3. – P. 275–293. doi : 10.1007/s11786-018-0345-5
9. Brown J. I. The average reliability of a graph / J. I. Brown, D. Cox, R. Ehrenborg // Discrete Applied Mathematics. – 2014. – Volume 177. – P. 19–33. DOI : 10.1016/j.dam.2014.05.048
10. Sahinoglu M. RBD tools using compression, decompression, hybrid techniques to code, decode, and compute reliability in simple and complex embedded systems / M. Sahinoglu, C. V. Ramamoorthy // IEEE Transactions on Instrumentation and Measurement. – 2005. – Volume 54, Issue 5. – P. 1789–1799. doi: 10.1109/TIM.2005.855103
11. Bennetts R. G. Analysis of Reliability Block Diagrams by Boolean Techniques / R.G. Bennetts // IEEE transactions on reliability. – 1982. – Volume R-31, Issue 2. – P. 159–166. doi : 10.1109/TR.1982.5221283
12. Catelani M. Reliability assessment for complex systems: A new approach based on RBD models / M. Catelani, L. Ciani, M. Venzi // Systems Engineering (ISSE) : 2015 IEEE International Symposium, Rome, 28–30 September 2015 : proceedings. – Rome : IEEE, 2015. – P. 286–290. DOI : 10.1109/SysEng.2015.7302771
13. A Practical Analytical Approach to Increase Confidence in PLD-Based Systems Safety Analysis / [A. V. da Silva Neto, L. F. Vismari, R. A. V. Gimenes et al.] // IEEE Systems Journal. – 2018. – Volume 12, Issue 4. – P. 3473–3484. doi : 10.1109/JSYST.2017.2726178
14. Redundancy allocation for series-parallel warm-standby systems / [O. Tannous, L. Xing, P. Rui et al.] // Industrial Engineering and Engineering Management : 2011 IEEE International Conference, Singapore, 6–9 December 2011 : proceedings. – Singapore : IEEE, 2011. – P. 1261–1265, doi: 10.1109/IEEM.2011.6118118
15. Synthesis of Neurocontroller for Intellectualization Tasks of Process Control Systems / [T. Teslyuk, V. Teslyuk, P. Denysyuk et al.] // The Experience of Designing and Application of CAD Systems (CADSM) : 2019 IEEE 15th International Conference, Polyana, 26 February – 2 March 2019 : proceedings. – Lviv : IEEE, 2019. – P. 1–4. DOI : 10.1109/CADSM.2019.8779295
16. Catelani M. A new proposal for the analysis of safety instrumented systems / M. Catelani, L. Ciani, V. Luongo // Instrumentation and Measurement Technology : 2012 IEEE International Conference, Graz, 13–16 May 2012 : proceedings. – Graz : IEEE, 2012. – P. 1612–1616. DOI : 10.1109/I2MTC.2012.6229556
17. Method of Reliability Block Diagram Visualization and Automated Construction of Technical System Operability Condition / [Yu. Ya. Bobalo, M. M. Seniv, V. S. Yakovyna et al.] // Advances in Intelligent Systems and Computing III. – 2019. – Volume 871. – P. 599–610. DOI : 10.1007/978-3-030-01069-0_43
18. Technique of automated construction of states and transitions graph for the analysis of technical systems reliability / [Yu. Bobalo, V. Yakovyna, M. Seniv et al.] // Computer Science and Information Technologies CSIT-2018 : XIII International Scientific and Technical Conference, Lviv, 11–14 September 2018 : proceedings. – Lviv : IEEE, 2018. – P. 314–317. DOI : 10.1109/STC-CSIT.2018.8526698
19. Techniques of automated processing of Kolmogorov – Chapman differential equation system for reliability analysis of technical systems / [Yu. Bobalo, V. Yakovyna, M. Seniv et al.] // The Experience of Designing and Application of CAD Systems (CADSM) : 2019 IEEE 15th International Conference, Polyana, 26 February – 2 March 2019 : proceedings. – Lviv : IEEE, 2019. – P. 267–272. DOI : 10.1109/CADSM.2019.8779271
20. Seniv M. Software for visualization of reliability block diagram and automated formulation of operability conditions of technical systems / M. Seniv, V. Yakovyna, I. Symets // Perspective Technologies and Methods in MEMS design MEMSTECH'2018 : XIV International Conference, Lviv-Polyana, 18–22 April 2018 : proceedings. – Lviv : IEEE, 2018. – P. 191–195. DOI : 10.1109/MEMSTECH.2018.8365731

DOI 10.15588/1607-3274-2020-4-16