http://doi.org/10.35784/iapgos.913

SYNTHESIS OF SAFE BEHAVIOR ALGORITHMS OF RADIOELECTRONIC SYSTEMS FOR CRITICAL APPLICATIONS

Leonid Ozirkovskyy, Bohdan Volochiy, Mykhailo Zmysnyi, Oleksandr Shkiliuk
Lviv Polytechnic National University, Department of Theoretical Radio Engineering and Radio Measurement, Lviv, Ukraine

Abstract. *The safety of radio electronic systems for critical applications is traditionally ensured by inducing structural redundancy. This paper shows a developed technique for ensuring a required level of safety of such systems by inducing time and functional redundancy into its behavior algorithm. The defined safety characteristic is proposed for quantitative efficiency estimation of the induced redundancy. Presented in the article is the synthesis technique of safe behavior algorithms on the basis of safety characteristic minimization of increased values. The developed technique was tested through solving the synthesis problem of the behavior algorithm of the target detection radio electronic complex system.*

Keywords: safety, safety engineering, behavior algorithm, system design

SYNTEZA ALGORYTMÓW BEZPIECZNEGO POSTĘPOWANIA W SYSTEMACH RADIOELEKTRONICZNYCH DO ZASTOSOWAŃ W SYTUACJACH KRYTYCZNYCH

Streszczenie. *Bezpieczeństwo radiowych systemów elektronicznych używanych w sytuacjach krytycznych jest tradycyjnie zapewnione przez wprowadzenie redundancji strukturalnej. W pracy zaproponowano sposób zapewnienia określonego poziomu bezpieczeństwa radiowych systemów elektronicznych poprzez wprowadzenie redundancji czasowej i funkcjonalnej do algorytmu postępowania. Aby zmierzyć wydajność wprowadzonej redundancji, zaproponowano określone cechy bezpieczeństwa. Artykuł przedstawia metodę syntezy algorytmów bezpiecznego postępowania na podstawie minimalizacji wzrostu wartości określonych cech bezpieczeństwa. Opracowana metoda została przetestowana podczas rozwiązywania problemu syntezy algorytmu postępowania dla złożonego systemu radioelektronicznego przeznaczonego do wykrywania celów.*

Słowa kluczowe: bezpieczeństwo, inżynieria bezpieczeństwa, algorytm postępowania, projekt systemu

Introduction

This paper represents a problem of behavior algorithm design which is developed for a radio-electronic system and should provide a high level of functional safety. The radio electronic system, as a part of a complex technical system, performs a task of organizing control actions. As an example of such complex technical system there can be mentioned a nuclear power plant, and a radio electronic system as its information driven system [8].

The exploitation safety of a complex technical system depends on several factors which include the functional safety and reliability of the radio electronic system [9]. That is why we use in this paper the definition "radio electronic system for critical applications" – RESCA.

An effect of the disability of a complex technical system is an emergency, and one of the reasons of failure is a non-execution of the RESCA task.

Therefore, the RESCA is designed as hardware and software system with fault tolerance structure [6, 11]. It consists of different subsystems and each of them performs its objective function. A spare subsystem should be provided for each subsystem to ensure the fault tolerance of the RESCA. The spare subsystem performs its function in a different way than the main one.

The safety of the RESCA is determined by its fault-tolerant structure and robust behavior, as well as its functional behavior [5, 10]. Functional behavior is defined by an algorithm that specifies the conditions and sequence of actions performed by the RESCA subsystems and modules when performing its functions. The behavior algorithm (BA) is being developed at the stage of the RESCA system design [1, 14]. Synthesis methods of different types of algorithms are discussed in details in monographs [3, 7, 12]. However, the synthesis problem concerning safe algorithms in these works is not solved.

There are more peculiarities of the RESCA behavior algorithms– they are characterized both by a successful and an unsuccessful execution [16]. Examples of such BAs are algorithms of target searching and detection, algorithms of receiving, recording and transmitting telemetry data, algorithms of obtaining navigation data, and others. An unsuccessful execution of such BAs leads to emergencies.

Failures of BA performance are caused by operator errors, hardware and software faults, inaccurate input data, a malfunction of control and diagnostics, etc. To minimize the number of unsuccessful executions of the RESCA and, accordingly, to ensure the required level of exploitation safety, the RESCA is provided with some means of time [2] and functional redundancy [15].

Verification of the feasibility of inducing each type of redundancy should take place at the stage of the RESCA system design. A lack of such ability in modern design means forced safety testing only at the testing stage of a complex technical system [13]. However, such approach is not always acceptable due to presence of errors in the RESCA behavior algorithm, when the complex technical system falls into an emergency. Such cases can be often seen in recent years after the launches of Soyuz rocket with spacecraft Progress (2011, 2015, 2016, 2017, 2018), SpaceX spacecraft Dragon (2015), military missiles Bulava (2017) and Burevestnik (2019). The main reason, for almost all emergencies, were errors in the BA of the launcher units, control systems, navigation systems. And it is important to detect these errors at the system design stage. Fortunately, these accidents happened only with the cargo spacecrafts.

Nevertheless, all consequences of such accidents can be very significant – these are a potential death of the crew, a contamination of large areas with hazardous substances (rocket fuel, radioactive substances etc.), a risk of potential fall of explosive objects on residential buildings, etc.

Therefore, a lot of attention should be paid to the problem of synthesizing the safe behavior algorithms of the RESCA. For qualitative solving of such a problem, it is necessary to operate tools (models, methods, techniques and software) which adequately take into account all the features of the behavior algorithm of the RESCA. So, all of it allows to carry out the synthesis of BA via multivariate analysis for short period of time. It should be noted that in order to confirm the accuracy of the obtained results at the stage of system design, it is necessary to operate not one, but two methods (or more) of solving the problem of BA analysis, since fulfilled reliability simulation results will allow to reduce the volume and, accordingly, the development cost of the RESCA field tests.

1. Problem statement

In order to provide a required level of safety for the RESCA exploitation, it is necessary to minimize the probability of failure of its behavior algorithm (unsuccessful execution). This necessity demands the re-execution of critical functions in case of hardware failures, software malfunctions, or operator errors. The re-execution of certain functions within the execution time period of

the BA requires a time redundancy, which implies the re-execution of some set of the BA operational blocks [2]. Accordingly, additional cycles are induced in the BA, which should contain blocks for checking stochastic and deterministic conditions [16]. In the case of a stochastic condition, it is not possible to predict in advance how many times the cycle will be executed and, accordingly, the duration of the cycle. If the execution of a certain function in certain conditions does not lead to a positive result, then the BA should provide the possibility of transferring the execution of mentioned function from one subsystem to the other, which performs this function in a different way. It means that the successful execution of the BA is ensured by functional redundancy.

A characteristic feature of the RESCA is the limit value of the duration of each BA objective function. If this value exceeds the allowable maximum, the algorithm fails to complete its operation and the RESCA falls into a critical failure. Thus, the induction of time redundancy in the BA may not be acceptable by certain requirements of the duration. To minimize the probability of unsuccessful execution of the BA, and accordingly, an emergency of a complex technical system, it is necessary to determine the maximum number of repetitions of each cycle.

Switching to another operating block means switching the function execution to another RESCA subsystem, which performs this function in another way. If, after several repetitive cycles the RESCA fails to complete the task and the average cycle time exceeded the limit, then switching to another RESCA subsystem is done. If this subsystem is not able to perform its function, it switches to the next one. If all the functions provided in the AP are completed, then we have a successful execution. Otherwise, we have an unsuccessful execution so, accordingly, the complex technical system falls into an emergency.

Checking the induction efficiency of each of the redundancy varieties should be done at the stage of the RESCA system design and the following questions should be answered:

- for which BA functions should a repeatable cycle be induced?
- how many times should each cycle be executed?
- will the average duration of the behavior algorithm exceed the limit value?
- should another RESCA subsystem to perform a non-executed function be switched to?
- how will the induction of redundancy impact the safety of the RESCA exploitation?

Therefore, at the system design stage, designers should have some set of models, methods and techniques that will provide the efficiency estimation of the RESCA, depending on the tactical and technical characteristics of hardware, indexes of reliability, different number of repeatable cycles of behavior algorithm, an operator's reaction rate, etc.

2. Method of synthesis of safe behavior algorithms

As it was mentioned above, for the synthesis of the safe BA, it is necessary to induce time and functional redundancies, and to determine their effect on the performance of the algorithm, since the efficiency and safety indexes of the BA are closely linked. This is explained by the fact that during unsuccessful completion, the BA includes successful executions and unsuccessful executions (emergency states).

Let's consider a model of all possible options for the implementation of BA while the RESCA operation is in the form of a graph of states and transitions. To do this, one has to assume that the RESCA is supposedly monitored, and the same BA is run many times. As a result of the BA execution, the RESCA gets into different states. States of current execution are defined by determined factors that operate the BA execution process (number of operating blocks, duration of operation, number of cycles, number of paths, etc.), and the states of successful execution and the emergency states are determined by random factors that lead to the failure (failure of subsystems, operator errors) or stoppage of the BA (average duration (T_{avg}) of BA exceeds the limit value of duration (T_{lim}), external interference, etc.).

Thus, there are three groups of states:

- the states from the first group reflect the process of transition from one operating unit of the BA to another in the process of performing the objective function;
- the states from the second group are the successful completion of the BA;
- the states from the third group are the unsuccessful completion of the BA.

2.1. Characteristics of behavior algorithm exploitation safety

As a result of the execution of the behavior algorithm, the RESCA is in the first group of states for a certain period of time, the average value of which (T_{avg}) should not exceed the limit of the execution duration (T_{lim}) of the objective function. Then, depending on the external and internal factors, the RESCA changes its state to the second group or to the third group. If the RESCA stays in the first group of states over the limit value (T_{lim}), it cannot perform its task and the complex technical system falls into an emergency. If, under the $T_{avg} \leq T_{lim}$, the RESCA enters the second group of states, and it completes successfully its objective function. If the RESCA falls into the third group of states, there are two consequences for them:

- for $t \leq T_{lim}$, the RESCA can return to the first group of states as a result of re-execution of some part of the AP. Moreover, there are several transmissions into the third group of states from the first state group and back.
- for $t > T_{lim}$, the RESCA remains in the third state group because the BA falls into the state of unsuccessful execution of the task, so the RESCA causes no emergency at all.

The changes of the RESCA state to the first and the second groups are characterized by the probability of successful execution of the BA – $P_{sp}(t)$. This is the probability that the RESCA completes its task as a result of the execution of the BA. This probability is the sum of probabilities of being in k states of successful execution – $P_j(t)$:

$$P_{sp}(t) = \sum_{j=1}^{k} P_j(t), \tag{1}$$

The changes of the RESCA state to the third state group characterized by the probability of unsuccessful (emergency) execution of the BA – Q(t). This is the probability of an unsuccessful completion of the BA, so as a result RESCA does not perform its task.

$$Q(t) = \sum_{i=q}^{v} P_i(t), \tag{2}$$

where $q, \ldots, v$ – numbers of unsuccessful execution states.

However, the probability of unsuccessful completion of the task is an integral characteristic of the BA and does not allow to estimate how many times during T_{avg} time the RESCA falls into emergency state. Therefore, we propose to define a new feature of BA exploitation safety – frequency of falling into the emergency state:

$$w(t) = \frac{dQ(t)}{dt} = \sum_{q=0}^{z-1} \lambda_{z,z-q} \cdot P_{z-q}(t), \tag{3}$$

where $\lambda_{z,z-q}$ – intensity of transition to the z state of unsuccessful execution of the BA from the z-q state,

$P_{z-q}(t)$ – the probability that the RESCA stays in the z-q state,

Q(t) – the probability that the RESCA gets in critical failure state.

Obtained characteristics of BA exploitation safety have the following variants (Fig. 1).

If the RESCA does not have functional and structural redundancy, then the maximum of the safety characteristic w (t) will be at time t = 0 and the characteristic will decrease exponentially.

When redundancy is presented then frequency of falling into emergency state *w(t)* at time $t = 0$ can take the value ≥ 0 and further increase to some maximum value, and later decrease

exponentially to zero. The more redundancy (temporal and / or functional) is induced into the BA, the lower is the value of the w(t) maximum and the further to the right it moves. In other words, with an increase of redundancy, the frequency of falling into an emergency state decreases.

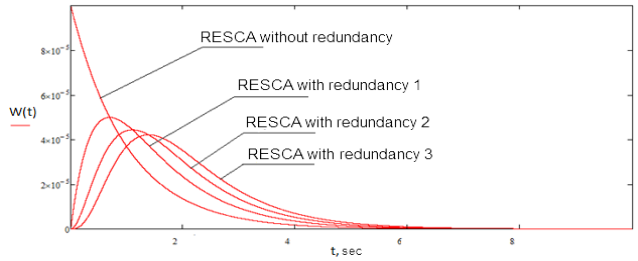

Fig. 1. Characteristics of BA exploitation safety

Figure 2 shows the relationship between safety characteristics and the RESCA efficiency.

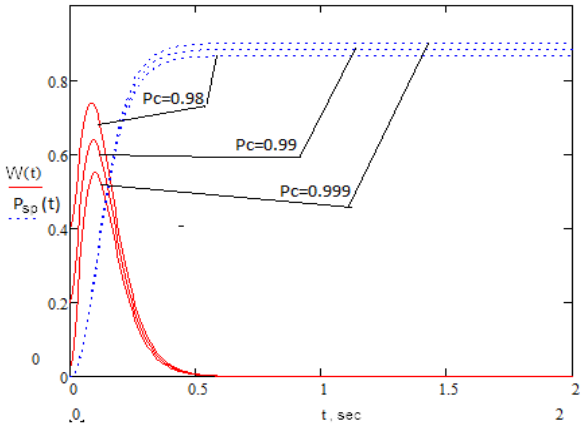

Fig. 2. Relationship between safety characteristics and the RESCA efficiency

Thus, the essence of the procedure for the synthesis of safe behavior algorithms for the RESCA is to minimize the maximum value of the frequency of falling into emergency state under the following restrictive conditions:

- a minimum permissible threshold for the probability of successful execution of the BA – $P_{lim}(t)$,
- a limited value of probability of unsuccessful execution of BA – Q(t),
- a limited value of the permissible average value of the execution duration of the BA – T_{lim}.

2.2. Technique of safe behavior algorithm synthesis

The technique of safe behavior algorithm synthesis consists of a set of methods and models presented by the scheme in Fig. 3. This technique consists of nine stages. Its essence is that at the first stage time redundancy is heuristically introduced into critical function of the BA of the RESCA to achieve successful completion. Practically its use is realized in a cyclic re-execution of certain operating blocks (functions). Such cycles of repetition of certain functions in the AP can be several dozens. Moreover, there is a certain contradiction: on the one hand it is impossible to determine the required number of repetitions, and on the other hand it is impossible to allow a loop on a certain function (causes too long cycle duration). If, after a certain number of repetitions, the subsystem is unable to complete the task, then its task is performed with some probability of successful execution by another subsystem, if it is possible. This case induces functional redundancy.

Functional redundancy is displayed in the BA by the induction of stochastic check blocks. If, under certain conditions of application, the BA with the induction of time and functional redundancy in the RESCA cannot complete the task with a given probability for a given time, the BA fails. Thus, in the first step, additional re-execution cycles (time redundancy) and stochastic checking blocks of functional redundancy need to be induced in the initial version of the RESCA behavior algorithm to provide a given level of safety for the RESCA exploitation. As a result, we get a new version of the BA for which it is necessary to select the parameters of time and functional redundancy, and operating modes of equipment and hardware to provide the necessary value of probability of execution and the average value of the execution duration of the RESCA task.

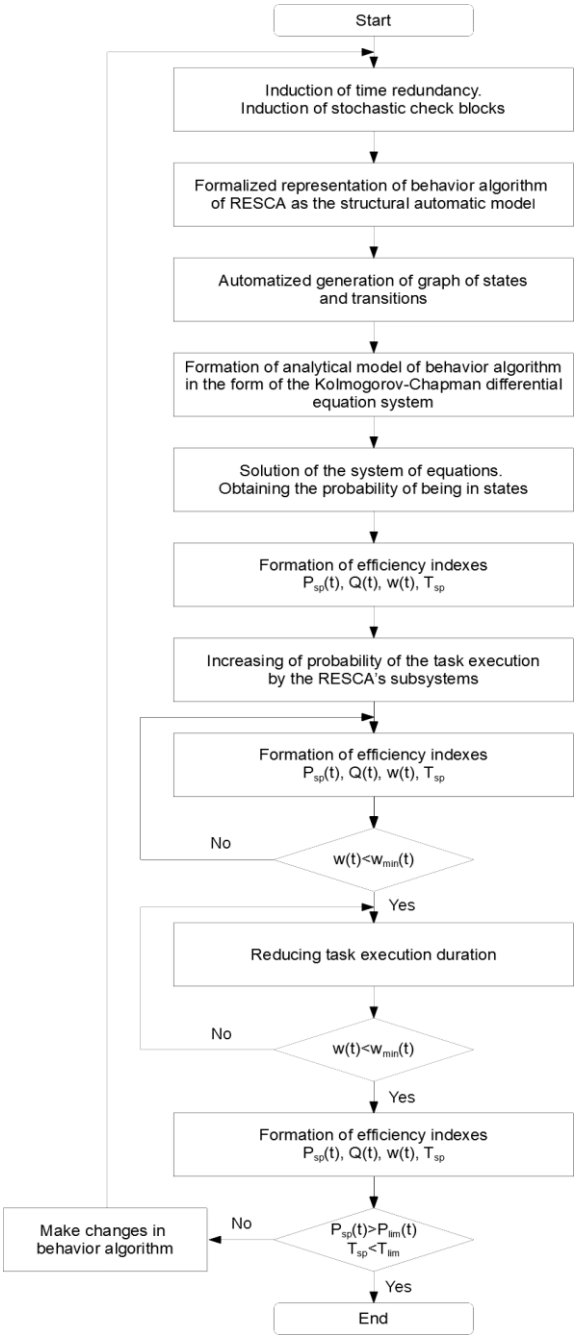

Fig. 3. Flowchart of the method of safe behavior algorithms synthesis

The second and the third steps of the technique provide a representation of the BA model in the form of a graph of states and transitions. An effective tool for such representation is improved space state method [4]. This method allows to obtain a complete state space, which includes all three of the mentioned above groups of states, it gives an ability to adequately reflect all variants of application of the RESCA behavior algorithm during the task execution.

In the second stage it is necessary to develop a structural-automatic model (SAM) [4] of the behavior algorithm. The peculiarity of such SAM is the presence of two types of check blocks (stochastic and deterministic) and one base event – the current execution of the operating block. At the third stage the model is automatically built in the form of a graph of states and transitions using the ASNA software [17].

At the fourth stage, an analytical model is formed in the form of Kolmogorov – Chapman differential equation system. This equation system is formed automatically using ASNA software. The solution of the differential equation system in the fifth stage will result in the distribution of probabilities of being in each state. From the resulting distribution, formulas are composed to determine the efficiency indexes of the BA as the sum of the probabilities of being in the respective states (the sixth step).

The obtained indexes depend on the parameters of the BA (the probability of performing the RESCA subsystems function, the average performance time of the RESCA subsystems function the probability of failure-free performance of the RESCA subsystem, the number of re-execution cycles of the BA operational units, the probability of switching to an alternative subsystem, etc.). Changing these parameters, one need to find the minimum value of the frequency of falling into emergency state – $w_{min}(t)$ in the state of unsuccessful completion of the BA (steps 7, 8 and 9).

The limiting conditions in this case are the minimum permissible limit of the task execution probability– $P_{lim}(t)$, and the maximum permissible limit value of task execution duration – $T_{lim}(t)$. If the safety characteristic w(t) takes the minimum value, and the $P_{sp}(t) \geq P_{lim}(t)$ (t) and $T_{sp} \leq T_{lim}$, then the obtained BA is safe. If this condition is not fulfilled, it is necessary to make changes in the BA and repeat all the above steps. Thus, the problem of finding the minimum value of the characteristics maximum of the BA is solved through a repeated change of the input data, which depends on the probability of the BA successful completion and its average duration.

This problem is solved with the help of ASNA software [17], which constructs new graphs of states and transitions, forms the systems of Kolmogorov – Chapman differential equations, and solves them on the basis of the BA automatic structural model. Based on these solutions, the BA exploitation safety characteristics are calculated. The technique was tested through the algorithm for searching, detecting and capturing targets of the air monitoring radio electronic system.

3. Conclusion

The application of the exploitation safety characteristics w(t) of the behavior algorithm made it possible to assess the impact of the induced time redundancy on the efficiency indexes of the radio electronic systems for critical applications.

We can estimate the frequency of behavior algorithm falling into the states of unsuccessful execution. The developed technique, then, allows system designers to minimize the frequency of behavior algorithm failures performing multivariate analysis considering both the configuration of the behavior algorithm and the hardware parameters of subsystems of complex technical system.

References

[1] Alexander R., Alexander-Bown R., Kelly T.: Engineering Safety-Critical Complex Systems. Proceedings CoSMoS 2008: Workshop on Complex Systems Modelling and Simulation. Luniver Press, 2008, 33–62.

[2] Ayoob M.; Adi W.: Fault Detection and Correction in Processing AES Encryption Algorithm. Proceedings Sixth International Conference on Emerging Security Technologies (EST), Braunschweig, Germany, 2015, 7–12.

[3] Benoit A., Robert Y., Vivien F.: A Guide to Algorithm Design. Paradigms, Methods, and Complexity Analysis. CRC Press Published, 2013.

[4] Bobalo Yu., Volochiy B., Lozinsky O., Mandziy B., Ozirkovsky L., Fedasyuk D., Scherbovskikh S., Yakovina V.: Mathematical Models and Methods for Reliability Analysis of Radio. Electronic and Software Systems. Lviv Polytechnic National University, 2013. (in Ukrainian).

[5] Dubrova E.: Fault-Tolerant Design. Springer, New York 2013.

[6] Hobbs C.: Embedded Software Development for Safety-Critical Systems. Second edition. Routledge, 2019.

[7] Kleinberg, J., Tardos E.: Algorithm design. First edition, Pearson, 2005.

[8] Pietrantuono R., Russo S.: Introduction to Safety Critical Systems. Innovative Technologies for Dependable OTS-Based Critical Systems. Challenges and Achievements of the CRITICAL STEP Project, Springer, 2013.

[9] Rausand M.: Reliability of Safety-Critical Systems: Theory and Applications. Wiley-Blackwell, 2014.

[10] Saha S., Sadi M. S.: Synthesizing fault tolerant safety critical systems. Journal of Computers 9(8)/2014, 1809–1816.

[11] Shafei E., Moawad I., Sallam H., Mostafa A.: A Methodology for Safety Critical Software Systems Planning. Proceedings 7th WSEAS European Computing Conference (ECC '13), Dubrovnik, Croatia, 2013.

[12] Skiena S.: The Algorithm Design Manual. 2 ed., Springer, 2009.

[13] Trukhanov V. M.: Reliability of Technical Systems of Mobile Units Types at the Stage of Prototypes Design and Testing. Mashynostroenie, 2003. (In Russian).

[14] Van Beek T., Tomiyama T.: Requirements for Complex Systems Modeling. Proceedings 18th CIRP Design Conference - Design Synthesis. Enschede, Netherlands, 2008.

[15] Viktorov D.: Algorithm Providing Fault Tolerance of Onboard Computing Systems with Structural and Time Redundancy. Information and Control Systems 6(2011), 30–35. (in Russian).

[16] Volochiy B., Ozirkovskyi L., Shkiliuk O. Mashchak A.: Technique of Construction Models of Behavior Algorithms of Radio Electronic Complex System using the Scheme of Paths Method. International Journal of Computing 13(3)/2014, 183–190.

[17] Volochiy B., Mandziy B., Ozirkovskyi L.: Extending the features of software for reliability analysis of fault-tolerant systems. Computational Problems of Electrical Engineering 2(2)/2012, 113–121.

Ph.D. Leonid Ozirkovskyy
e-mail: l.ozirkovsky@gmail.com

Ph.D., Associate Professor of Theoretical Radio Engineering and Radio Measuring Department of Lviv Polytechnic National University.
Experience of teaching in higher education for is over 19 years. Author of 140 scientific publications including 2 monographs, 5 textbooks. He has prepared 3 Doctors of philosophy (PhD). Research interests include the development of methods and tools for modeling functional and reliability behavior of information systems, safety engineering, reliability engineering.

http://orcid.org/0000-0003-0012-2908

D.Sc. Bohdan Volochiy
e-mail: bvolochiy@ukr.net

D.Sc., Professor of Theoretical Radio Engineering and Radio Measuring Department of Lviv Polytechnic National University
Experience of teaching in higher education for is over 40 years. Author of 280 publications, including 4 monographs, 3 textbooks, 5 inventions. He has prepared 4 Doctors of philosophy (Ph.D.). Research interests are: theory and practice of system design of radio electronic information systems.

http://orcid.org/0000-0001-5230-9921

Ph.D. Mykhailo Zmysnyi
e-mail: zmysnyim@gmail.com

Ph.D., Senior Lecturer of Theoretical Radio Engineering and Radio Measuring Department of Lviv Polytechnic National University
He has been teaching in higher educational establishments for over 5 years. The author of over 30 scientific publications. Research interests include the development of methods and tools for modeling functional and reliability behavior of fault-tolerant systems with majority structure.

http://orcid.org/0000-0002-3384-6139

Ph.D. Oleksandr Shkiliuk
e-mail: oleksandr.p.shkiliuk@lpnu.ua

Ph.D., Senior Lecturer of Theoretical Radio Engineering and Radio Measuring Department of Lviv Polytechnic National University
He has been teaching in higher educational establishments for 5 years. The author of over 30 scientific publications. Research interests include the development of methods and tools for modeling functional and reliability behavior of algorithms of radio electronic critical systems.

http://orcid.org/0000-0001-9237-4808

otrzymano/received: 15.11.2019 *przyjęto do druku/accepted: 15.02.2020*