Cognitive-Impulse Model For Assessing Complex Technical Systems Survivability

Vladimir Vychuzhanin[1] [0000-0002-6302-1832], Nicolay Rudnichenko[1] [0000-0002-7343-8076], Natalia Shibaeva[1] [0000-0002-7869-9953], Yuriy Kondratenko[2] [0000-0001-7736-883X], Igor Gritsuk[3] [0000-0001-7065-6820]

[1]Odessa National Polytechnic University, Odessa, Ukraine
vint532@yandex.ua, nickolay.rud@gmail.com, nati.shibaeva@gmail.com
[2]Petro Mohyla Black Sea National University, Mykolayiv, Ukraine
y_kondrat2002@yahoo.com
[3]Kherson State Maritime Academy, Kherson, Ukraine
gritsuk_iv@ukr.net

Abstract. The article proposes to assess the system components failure risk during diagnostics based on the research methods analysis results and complex technical systems models, in order to ensure their survivability. The authors developed a method for assessing the survivability of systems based on cognitive-impulse modeling and interaction with a complex technical system, as with a digraph subjected to the impulse action of simulation modeling impulses. The created cognitive-impulse model allows us to evaluate the survivability, the degree of structural and functional damage to the complex technical system components and intercomponent connections. The developed cognitive-impulse model of survivability assessment has flexibility - the ability to use the method at any level of subsystem components failure risk assessment with their various configurations; adaptability - the ability to adapt to changes in the configuration of complex technical systems subsystems. The developed software allows to obtain both numerical and graphical results of various classes technical systems cognitive-impulse models studies and complexity. The developed cognitive-impulse model effects over a complex technical system makes it possible to establish emergency and pre-emergency conditions for its subsystems.

Keywords: complex technical system, survivability, cognitive impulse modeling, software

1 Introduction

The structure and functional properties of complex technical systems (CTS) are determined by numerous heterogeneous components interacting with each other, the number of which can reach thousands. In CTS design and operation, the reliability requirements for systems are constantly being tightened. This is associated with the failures risk, accidents and disasters arising from the technical systems operation. The disruption of such systems functioning is possible if the connectivity of their structures is broken. The system cannot perform its functions when there are no interactions between all or its vital components. From the point of view of the security concept, CTS must be studied from the standpoint of reliability, survivability.

2 Description of Problem

Among the models used to assess CTS survivability, we can distinguish: normative and descriptive approaches; probabilistic and deterministic models; fault trees; logical-probabilistic and analytical methods [1-8]. Methods associated with structured objects and fuzzy knowledge are used to assess the survivability of CTS. In practice, these methods are rarely found in their pure form. Therefore, one of the most common survivability assessment systems is combined systems (for example, combining normative and logical-probabilistic approaches). The most developed methods are logical-probabilistic models and fault trees. Another rapidly developing area is cognitive-simulation modeling [7,9,10].

CTS survivability assessment is impossible without taking into account the structural and functional relationships of their subsystems and components [11,12]. To do this, the CTS failure risk assessment should take into account the division of systems into subsystems (complexes, aggregates), components (components and parts). As part of the subsystem, structurally and functionally completed components of the system can be considered, the interaction of which should ensure its reliable operation. Each CTS component is associated with other components in a specific way. Therefore, when assessing the CTS subsystems component failures risk it is necessary to identify their interconnections and interactions, conducting a structural systems analysis. When researching the CTS model, it is necessary to establish the features of the system's functioning in various operating conditions and reduce the failure risk of its subsystems and components. If we consider the problem of assessing the failures risk in diagnosing CTS state from risk management theory view point [13,14], then the corresponding model for assessing the survivability of a system should contain the main components that affect its quality and functioning efficiency. Such an approach in studies without detailing CTS, processes and phenomena occurring in them is a system synthesis [15,16].

In order to ensure CTS survivability under various operating conditions, to successfully investigate systems reliability during their transitions between different state variants, further development of methods for assessing CTS survivability, taking into account the structural, functional interactions and interactions CTS subsystems and components [17- 20] is necessary. It is actual to develop methods and models, new solutions for CTS survivability assessment informatization to ensure: flexibility - the ability to use methods at any level to assess the subsystems components failure risk with their various configurations; adaptability - methods must have the ability to adapt when changing CTS subsystems configuration.

Thus, the solution of the systems ensuring the survivability problem upon receipt of information about failures in their components and subsystems remains relevant in the technical condition diagnosis.

3 Principles of constructing a cognitive-impulse model for assessing complex technical systems survivability

In order to ensure CTS survivability, taking into account the structural and functional interactions and interactions of components and subsystems, it is proposed to use system synthesis based on a probabilistic-deterministic model that describes impulse propagation effects on system components. CTS research experience shows that at the stage of assessing the survivability of its components and subsystems, it is advisable to represent them as an oriented graph, the vertices of which are components (subsystems) [21] that have certain properties, and describe the interaction of components (subsystems) using oriented graph edges. The survivability measure in this case is the minimum number of CTS components (vertex connectivity [22]) or connections (edge connectivity [22]), the failure of which under the influence of external influences leads to a breakdown in the system structure connectivity.The proposed method uses complex cognitive-impulse modeling and interacts with CTS, as with a digraph subjected to impulse effects of different modeling and distribution modeling simulation impulses. The essence of this modeling is that at one of oriented graph vertices a certain change is specified. This vertex actualizes the entire indicators system, i.e. peaks associated with it. Passing along the vertices and edges of the oriented graph, the impulse changes their states and changes itself (depending on the model, the impulse impact modulus can change). In assessing CTS structural threats, the effect of the unchanging striking modeling impulse (SMI) is used, when propagating through a oriented graph that modifies the state of individual components (subsystems). Functional threats are evaluated using a varying diagnostic modeling impulse (DMI), which propagates through the oriented graph and does not affect the vertices.

The method uses the concepts of threats and risks. The threat is a dimensionless off-scale assessment component vulnerability in terms system structural survivability as a whole. CTS survivability is considered as a combination of two main factors - CTS vulnerability as a complex structure and the likelihood of CTS certain components failure as a result of external or internal adverse effects. Threat analysis reveals vulnerabilities - “weaknesses” in the CTS structure. The basic concepts in the proposed method are the concepts of potential structural and functional threats. A potential structural threat is a component position assessment in the CTS structure and its role in the system topology. A functional threat also takes into account the position and role of the component in the system, but from the point of view of potential changes in the system’s operability, where operability means the probability that the system will successfully complete tasks after exposure to adverse and damaging factors. CTS is considered to be affected by external influences. An increase in the load can be observed not only in those system components that are influenced by external influences, but also in the components interacting (connected) with them. It should be noted that we are talking only about loads that are not normative, i.e. taken into account in the system design and operation. Short-term and powerful external (pulsed) impacts can instantly and significantly reduce the reliability indicators of individual components and the entire system as a whole, but CTS is able to maintain its operability.

4 Model of striking effects spread on CTS

In the CTS digraph, vertices correspond to components (complexes, systems, subsystems, elements), and directed arcs correspond to one of the types of intercomponent connections (transfer of matter, information, or energy). Many vertices (components) of a oriented graph V – $(V=\{v_i\}, i=\overline{1,N})$. Pair vertices v_i – oriented graph edge – component interconnection (CI). CI set $A-(A=\{a_j\}, j=\overline{1,M})$. SMI impacts are determined by the pulse vector $imp_{i,j}(t)$ for discrete time t = 0,1,2,3... Each component (CI) characterized by specific properties, the combination of which determine the state of the components (CI). The qualitative state of the CTS components is expressed by the state functional

$$v_i = (F_{v_i}, a_{ji}, a_{ij}, H_m^{v_i}(t), K_{v_i}), \tag{1}$$

where F_{v_i} – current component performance; a_{ji}, a_{ij} – quality state of incoming and outgoing for a component CI; K_{v_i} – component structural striking degree coefficient (SSDC); $H_m^{v_i}(t)$ – amplitude change SMI coefficient, passing through the CTS component

$$H_m^{v_i}(t) = \frac{imp_{m_{i+1,j}}(t)}{imp_{m_{i,j}}(t)}, \tag{2}$$

where $imp_{m_{i,j}}(t)$, $imp_{m_{i+1,j}}(t)$ – SMI amplitude value at the CTS component input and output

$$K_{v_i} = H_m^{v_i}(t) \cdot \frac{w_{v_i}(t+1)}{w_{v_i}(t)} \tag{3}$$

where $w_{v_i}(t)$, $w_{v_i}(t+1)$ - the value of the component (vertex) weight at time points t, $t+1$ as a result of exposure to SMI.

Qualitative state of CI CTS is expressed by the state functional

$$a_j = (F_{a_j}, v_{ij}, v_{ji}, H_m^{a_j}(t), K_{a_j}), \tag{4}$$

where F_{a_j} – CI current performance; $v_{i,j}, v_{j,i}$ – quality condition of components at the CI beginning and end; K_{a_j} – SSDC CI strike; $H_m^{a_j}(t)$ – SMI change amplitude coefficient passing through CI.

$$H_m^{a_j}(t) = \frac{imp_{m_{i,j+1}}(t)}{imp_{m_{i,j}}(t)}, \quad (5)$$

where $imp_{m_{i,j}}(t)$, $imp_{m_{i,j+1}}(t)$ – amplitude value SMI at input and output CI.

$$K_{a_j} = H_m^{a_j}(t) \cdot \frac{w_{a_{i,j}}(t+1)}{w_{a_{i.j}}(t)}, \quad (6)$$

where $w_{a_{i,j}}(t)$, $w_{a_{i,j}}(t+1)$ - weight value CI (edges) at time points t, $t+1$ as a result SMI exposure.

The structural vulnerability vertices (edges) structural damage degree coefficients (SVC) reflect the level of threat assessment of the system, allowing us to rank the CTS components according to the degree of structural significance and highlight the least reliable.

SMI impact on the vertex (edge) of the graph at a discrete time t determined by

$$1 - imp_{i(i,j)}(t) = \frac{w_{v_i(a_{i,j})}(t)}{w_{v_i(a_{i,j})}(t-1) \cdot K_{v_i} \cdot K_{a_j}}, \quad (7)$$

where $imp_{i(j)}(t)$ – impulse vector for vertex (edge) with number i (i, j); $w_{v_i(a_{i,j})}(t)$, $w_{v_i(a_{i,j})}(t-1)$ – value edge (vertex) weight at a time t and at the previous moment in time $t-1$. Oriented graph vertex weight $w_{v_i}(t)$ - the magnitude of its reliability for the top v_i vertex. Edge weight $w_{a_{i,j}}(t)$ is determined by a numerical value from 0 to 1 and is equal to the proportion of SMI from v_i to v_j. When DMI passes through the CTS components at time t, the impulse actions are determined for the connected components:

- sequentially (to the top v_i), t+1 (after passing v_j), t+2 (after going through the v_i and v_j)

$$imp(t+2) = imp(t+1) \cdot v_j = imp(t) \cdot v_i \cdot v_j. \quad (8)$$

- in parallel (before passing v_i and v_j), t+1 (passing v_i and v_j), t+2 (after passing the junction point of the oriented graph)

$$imp(t+2) = imp(t) \cdot v_i + imp(t) \cdot v_j = imp(t) \cdot (v_i + v_j). \quad (9)$$

Similarly, one can obtain relations for more complex series-parallel structures.

Representation of CTS in the form of a oriented graph and formalization of the external influence on the system as an impulse action (1) - (6) defines a model of the distribution of damaging effects across the system. Consider the impulse effect on the CTS, the components of which have survivability and bonds equal to unity, using SMI. The impact simulates the defeat of the object number i by a force impact 1.0 and is equivalent to the complete defeat of the component and its failure. In a single system, struck by a single impulse, the connections between the vertices of the oriented graph are also single, so the SMI will propagate through the CTS until it disables all available components. This is a case of the "worst case scenario" for a single hit. Consequently, the criticality of an element (and with it the degree of damage to the system) can be estimated by damage volume it causes to the CTS. As an example of such an impact, consider a oriented graph (fig. 1). We sequentially act on each of its vertices with a single impulse and follow the propagation of the impulse along the oriented graph. We believe that in Fig. 1, the final phase of the impact of a single SMI falls on the peak of V9.

Consider the scenario of the defeat of a system unit by a single impulse (assuming that the vertices of the oriented graph have unit conductivity). The pulse will propagate through the system until it disables all available components. Hence, the criticality of the edge (and with it the vulnerability of the system) can be estimated by the extent to which the damage to the CTS as a whole causes the damage. We sequentially act on each edge of the oriented graph with a single SMI and trace the propagation of the impulse along the oriented graph. For example, Fig. 1 shows the final phase of an impulse propagation in the event of a connection failure E8, 10 (failed digraph connections are indicated by a dotted line). Peak and link structural vulnerability assessments provide, as a first approximation, a weight estimate of the vertex or link significance to ensure CTS survivability. Numerically, such an estimate is the greater, the more vulnerable system component.

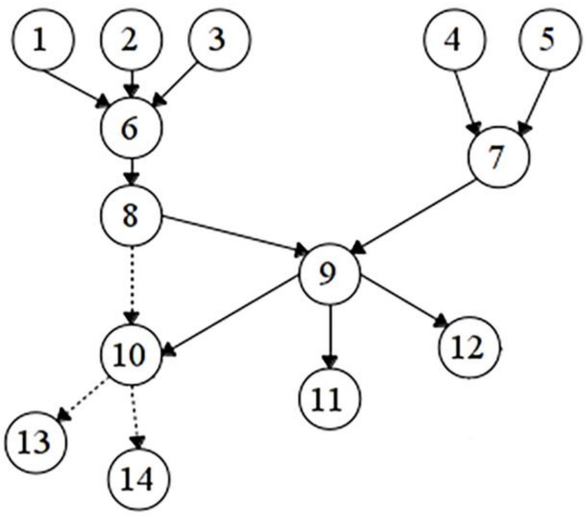

Fig.1. CTS oriented graph

As an example, lets study a numerical simulation of the oriented graph vertices (edges) strike shown in Fig. 1. As a result, we obtained the following SSDC values of oriented graph vertices.

Table 1. The coefficients of the oriented graph vertices damage structural degree

V1	V2	V3	V4	V5	V6	V7
0.71	0.72	0.7	0.6	0.6	0.64	0.58
V8	V9	V10	V11	V12	V13	V14
0.58	0.52	0.23	0.06	0.2	0.06	0.06

In accordance with the obtained results (Table 1) in the oriented graph under consideration, the vertices V1, V2, V3 are structurally important, V4, V5, V6, V7, V8, V9 are less structurally significant, the others are V10, V12, V11, V13, V14.

It should be noted that a change in a single DMI in the case of passing through the oriented graph vertices, provided that vertices weights are numerically equal to the assessment of their performance, will be numerically equal to the overall performance of the system. Therefore, modeling the passage through CTS of a single DMI allows us to evaluate system performance according to the above criteria.

5 Cognitive impulse simulation software

The following tools were chosen to simulate DMI effect on the system: presentation of the original models in JSON format; high level Python programming language; graphviz graph visualization tools; make utilities; Debian GNU / Linux distribution tools. The simulation was carried out on the basis of the Debian GNU / Linux distribution, with widespread use of shell software (makefiles, bash scripting language).

The utility uses special make-files, which indicate the dependencies of the files from each other and the rules for satisfying them. Based on the information about the time of each file last change, make determines and runs the necessary programs. Technologically, the modeling process using the make-utility looked as follows: the original model was formulated in terms of the JSON format, then transferred to the Python program input. Program result was an array of tables in *.csv format and files in *.dot language format. Then the array of dot-files using the make utility was batch transferred to the input of the graphviz program, the result of which was an array of the corresponding diagrams in *.png format. Analysis of the results was carried out by means of Calc Libre Office. To simulate the damaging and simulating impulses, two separate scripts were created, each of which contains a set of common functions. Both programs include the main part, auxiliary functions and the SMI or DMI simulation module, depending on the program purpose. The set of auxiliary functions is the same for both programs and contains the following functions: GetJson - analyzes the JSON file and builds a CIM based on it; DotFileDot - generates a dot-file with damaged nodes and passing pulses; DotFile - generates a sequence of dot-files for graphviz; Meet_In_Other - performs auxiliary functions (search for a node intersecting with vectors of other nodes); TablesToCsv - saves the results of the program to an array of csv tables. The HitPulse and DiagPulse routines, respectively, are used to model the effects of DMI. Programs are organized in such a way that they can be called both on their own (in stand alone mode) and as part of more complex software systems (in the form of plug-in libraries of the Python language).

Further processing of the results is performed using the GNU make utility and the grpaphviz visualization utility. A file in the dot language format is a text description of the graph in which the design of the nodes and the links between them is specified. To facilitate batch processing of the received dot-files, a make-file was used. Such a scheme made it possible to generate as all files at once (using the make directive either make all), or separately (make_st for SMI, make_fun for DMI), as well as to clean the directory before starting a new modeling cycle (make clean). In addition to the GNU make utility, GNU core utilities were widely used in modeling, mainly for converting csv tables to a form convenient for visualization using Calc Libre Office tools. The main uses were the GNU awk table data processing language and the paste utility included in the GNU coreutils package.

The logical concept of the developed software in Python is based on a microservice architecture with the goal of providing scalability and distribution of computational operations for complex systems with a large number of elements and interconnections. The user interface was developed by means of the QT Designer graphical environment with the connection of the pyQt library; it is a *.ui format file that stores the hierarchical structure of widget placement for processing user actions (data input, initialization of computational processes and graph construction) in a declarative form. The program code is implemented using the PyCharm IDE, based on 6 separate classes: main (the entry point to the program, contains all external dependencies and binds the structure of other classes), process (performs the functions of processing input data from a json file and importing them into dictionary structures), controller (provides the logic for processing user interaction with interface elements), calculate (implements the calculation functions for performing the simulation), visualize (implements the rendering and saving of the simulation results in the form of graph), logger (logs user operations performed in the program and simulation results into separate text files in *.log format). The diagram of the main options for using the simulation system is shown in Fig. 2. The user can import data by starting the json file parsing procedure. Set parameters for performing calculations, in particular, restrictions on the number of elements, degree of nesting, or calculation time, as well as parameters for generating graph images (image resolution, font type and size, color of edges and vertices). Set parameters for performing calculations for a given system structure for functional and structural damage, viewing calculation results in the form of diagrams and tables, saving and loading a model (data serialization and deserialization), saving generated action logs during the program operation, saving graph models in the process modeling, as well as exporting data in tabular form in *.xls or *.csv formats.

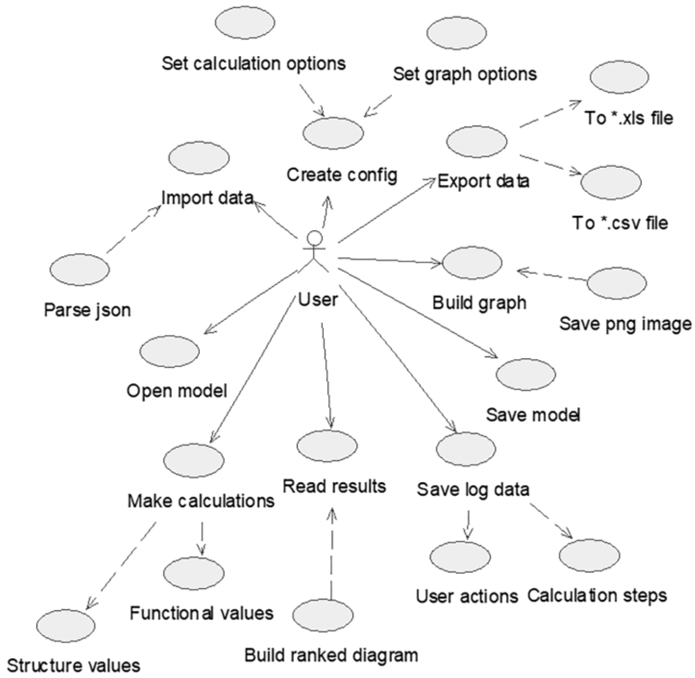

Fig.2. Simulation system use cases

As an illustration CMI software use is shown in Fig. 3. It shows the CTS (communication network of some control system), depicted as a oriented graph, and Fig. 4 shows modeling signal passage. The network should ensure the passage of the modeling signal from V6 to V2. A signal is sent to V6, which should reach V2. The main event S is the signal passing from the V6 to the V2. Intermediate events S_i, i = {1,2,3,4,5,6} - signal not passing to the V2.

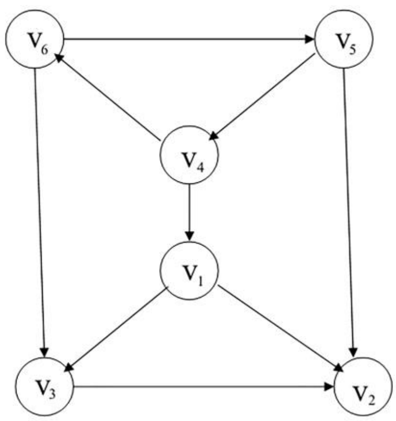

Fig.3. Oriented graph of the communication network management system

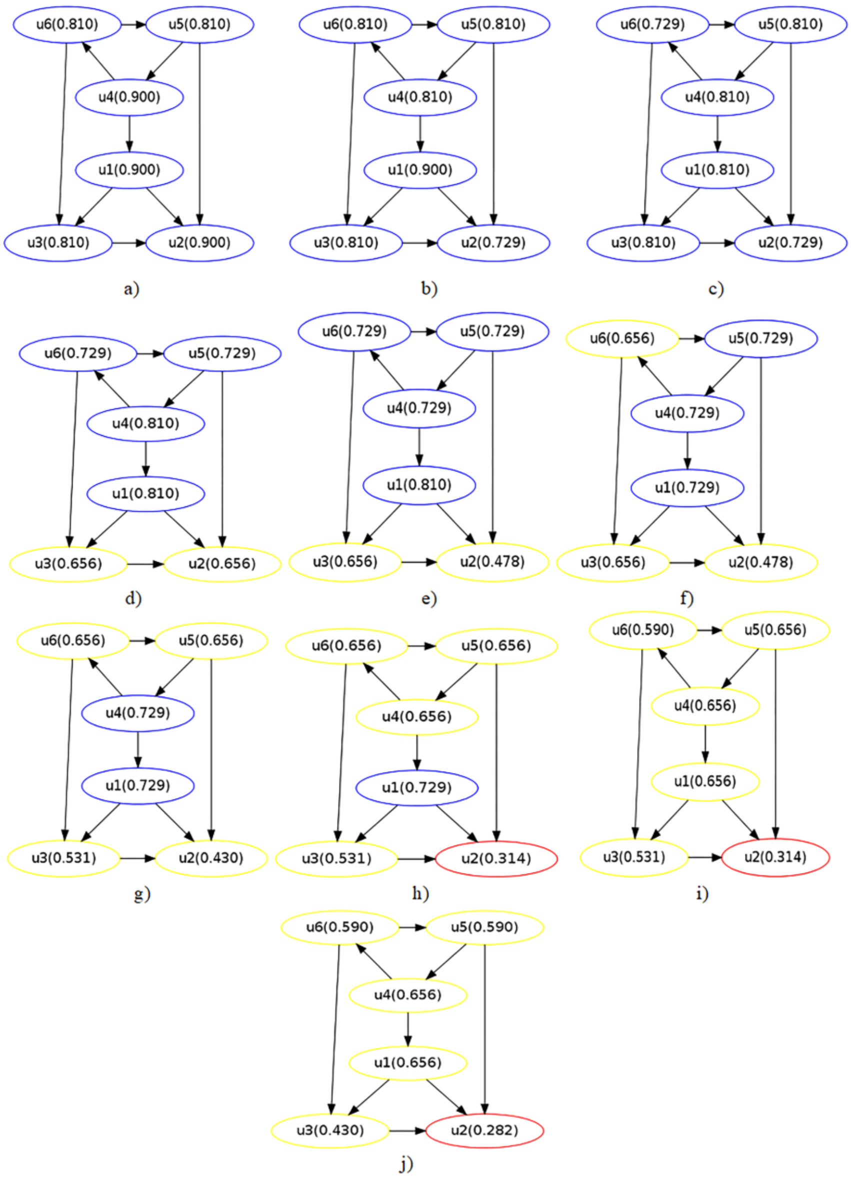

Fig.4. Modulating pulse movement

Thus, at a basic level, the structural threat to survivability is assessed by the sequential impact of SMI on each oriented graph vertices and then SSDC is calculated. The functional threat to the survivability of each oriented graph vertex is estimated as total DMI value ratio that passed through a CTS and the total DMI value during the sequential each vertex model failure. The ratio of the obtained values allows to obtain the CTS functional degree affection coefficient.

6 Cognitive impulse CTS survival assessment research

To conduct a CIM of assessing CTS survivability, an air conditioning system (ACS) operating in the “summer” mode of operation was selected as an object (Fig. 5). The system consists of units: Cm - compressor; E - electric motor; K is the capacitor; V is the evaporator; R1, R2 - control valves; N is the pump; Cn - control system; A - air cooler. A distinctive feature of the model under consideration is the non-hierarchical structure of the connections of its aggregates - the presence of a closed loop "Cn - K - R1 - V", as well as the presence of two types of connections - resource (solid lines) and information (dashed).

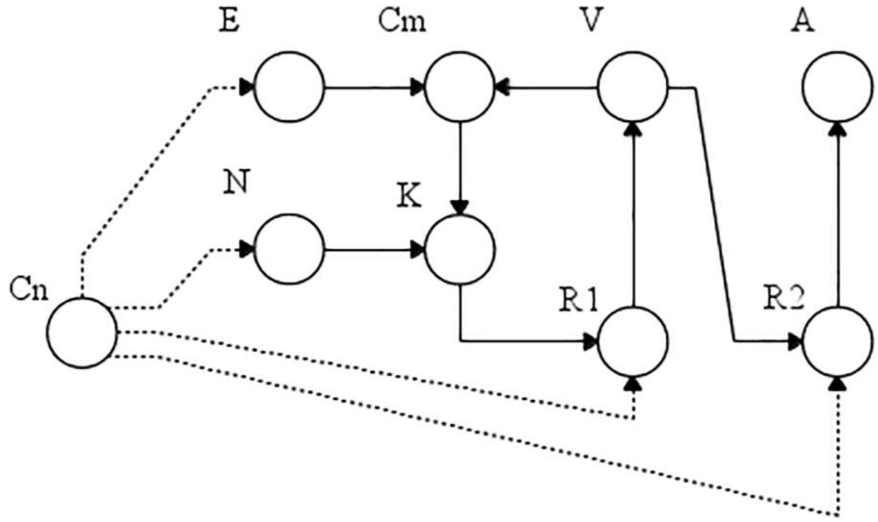

Fig. 5. ACS oriented graph in summer mode

To calculate SSDC, the vertices of the cognitive-impulse ACS model were sequentially exposed to SMI. Changes in vertex states for the initial and final stages of SMI propagation over the oriented graph are shown in Fig. 6.7 (disabled units are indicated by a dotted line). For aggregate Cm SVC ks = 0.667 (Table 2).

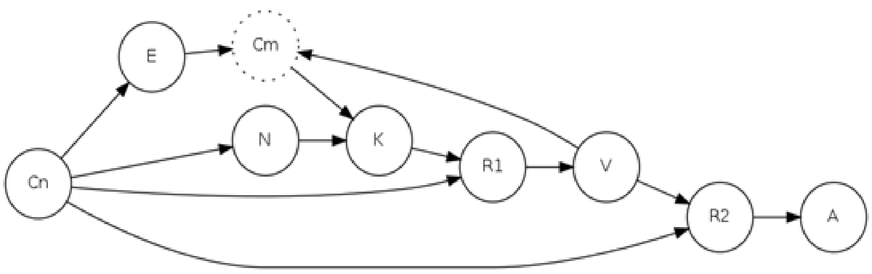

Fig. 6. Initial stage of SMI exposure to Cn CIM ACS

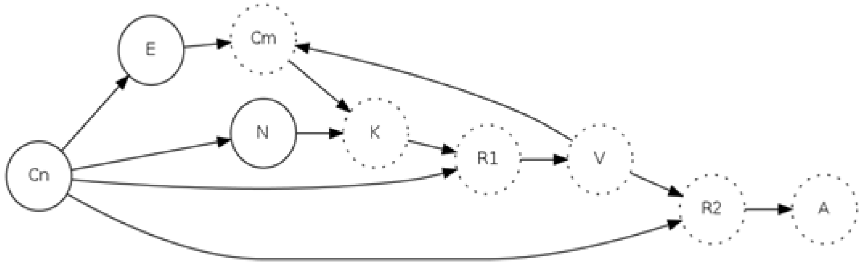

Fig. 7. The final stage of SMI exposure to Cn CIM ACS

Table 2. The coefficients of the ACS aggregates destruction structural degree

Unit	k_{v_i}
A	0.111
Cm	0.667
Cn	1.000
E	0.778
K	0.667
N	0.778
R1	0.667
R2	0.222
V	0.667

Based on the calculated SSDC values and the reliability of the units, the values of their structural failure risk were calculated (Table 3).

Table 3. ACS aggregate ranking by structural risk level of ACS unit's failure

Unit	r_s	Unit	r_s
Cm	163,55	R1	9,13
K	76,11	Cn	3,15
N	46,99	R2	3,04
E	32,11	A	2,28
V	18,26		

In order to assess functional threats (FT) and failure risk (FR) of aggregates within CIM, we expose the ACS digraph to DMI and analyze its distribution. Let us assume the conditional efficiency of all model vertices equal Fa = 0.9. Having ranked the results on the risk of failures (Table 4), we establish the most important ACS units for functioning. The system operability is dependent on the state of the units E and Cm.

Table 4. ACS aggregate ranking by FT and FR

Unit	Θ_f	$r_f \cdot 10^{-3}$	Unit	Θ_f	$r_f \cdot 10^{-3}$
Cm	0,99999998	0,24533	V	0,34072216	0,00933
E	0,99999998	0,04129	R2	0,00101010	0,00001
N	0,67367395	0,0407	Cn	0,00101010	0,00001
K	0,67367395	0,07691	A	0,00101010	0,00002
R1	0,34072216	0,00466			

In conducting the ACS studies, eighteen scenarios and one hundred thirty separate state diagrams were considered for its nine units. As a result of cognitive-impulse modeling, the values of structural and functional threats and risks were obtained, vulnerabilities in the CTS were identified. The study of ACS cognitive-simulation modeling allowed us to determine structural and functional vulnerabilities and determine the “weight” and “contribution” of aggregates to the overall survivability of the system, taking into account its layout and structure. The introduction of a reliability indicator allows us to refine the ranking of aggregates by risk and obtain data that can be used in the operation of the system and at the stage of its maintenance, modernization and design. In accordance with the described impulse action on the oriented graph, one can introduce various criteria (signs) for the system to reach the limit state. It should be considered that the CTS is in a critical state if the reliability of one or more of the most significant system components is below a certain acceptable level (critical level of element reliability). Such a criterion unambiguously separates the CTS component subcritical and supercritical state.

7 Conclusion

The proposed method for assessing CTS survivability is based on integrated cognitive-impulse modeling and interaction with the system, as with a oriented graph subjected to impulse action of different modeling impacts of different propagation patterns and changes. The created cognitive-impulse model makes it possible to evaluate the survivability of components, intercomponent connections and CTS as a whole, to determine the degree of structural and functional damage to CTS components and intercomponent connections. The developed cognitive-impulse model of survivability assessment has flexibility - the ability to use the method at any level of assessing the subsystems components failure risk with their various configurations; adaptability - the ability to adapt to changes in the configuration of CTS sub-systems. The developed software of the cognitive-impulse model for assessing CTS survivability allows one to obtain both numerical and graphic results models studies systems of various classes and complexity.

Reference

1. Atamanyuk, I., Kondratenko, Y.: Computer's Analysis Method and Reliability Assessment of Fault-Tolerance Operation of Information Systems. In: ICT in Education, Research and Industrial Applications: Integration, Harmonization and Knowledge Transfer. Proceedings of the 11th International Conference ICTERI-2015 // S.Batsakis et al. (Eds.), CEUR-WS, Vol. 1356, pp. 507-522 (2015)
2. Atamanyuk, I., Shebanin, V., Kondratenko, Y., Volosyuk, Y., Sheptylevskyi, O., Atamaniuk, V.: Predictive Control of Electrical Equipment Reliability on the Basis of the Non-linear Canonical Model of a Vector Random Sequence. In: Proceedings of the International Conference on Modern Electrical and Energy Systems, MEES 2019, 8896569, pp. 130-133 (2019)
3. Hakobyan, A., Aldemir, T., Denning, R., Dunagan, S., et al.: Dynamic generation of accident progression event trees. In: Nucl. Eng. Des., №238, pp. 3457–3467 (2008). doi: 10.1016/j.nucengdes.2008.08.005
4. Ibánez, L., Hortal, J., Queral, C., Gómez-Magán, J., Sánchez-Perea, M., Fernández, I., et al.: Application of the integrated safety assessment methodology to safety margins. Dynamic event trees, damage domains, and risk assessment. In:Reliab. Eng. Syst. Saf., №147, pp. 170–193 (2015). doi: 10.1016/j.ress.2015.05.016
5. Mandelli, D., Yilmaz, A., Aldemir, T., Metzroth, K., and Denning, R.: Scenario clustering and dynamic probabilistic risk assessment. In: Reliab. Eng. Syst. Saf., № 115, pp. 146–160 (2013). doi: 10.1016/j.ress.2013.02.013
6. Ryabinin I.A.: Nadezhnost' i bezopasnost' strukturno-slozhnykh sistem. SPb.: Politekhnika, (2000)
7. Vychuzhanin, V.V., Rudnichenko N.D.: Metody informatsionnykh tekhnologiy v diagnostike sostoyaniya slozhnykh tekhnicheskikh sistem. Monografiya, Odessa (2019)
8. Tsvetkov, V.YA.: Slozhnyye tekhnicheskiye sistemy. In:Obrazovatel'nyye resursy i tekhnologii, №3(20), pp. 86-92 (2017)
9. Rudnichenko, N. Vichuzhanin, V.: Information cognitive model of technological interdependence of marine technical recourses. In: Informatics and mathematical methods in simulation, Vol.3, №3, pp. 240-247 (2013)
10. Vychuzhanin, V. Rudnichenko, N., Zenkin, E.: Cognitive Model of the Internal Combustion Engine. In: SAE International, 2018-01-1738, pp. 1-10 (2018). doi:10.4271/2018-01-1738
11. Vychuzhanin, V. Rudnichenko, N., Sagova, Z., et al.: Analysis and structuring diagnostic large volume data of technical condition of complex equipment in transport. In: Machine Modelling and Simulations (2019). doi:10.1088/1757-899X/776/1/012049
12. Dorozhko, I. V., Tarasov, A. G., Baranovskiy A. M.: Otsenka nadezhnosti strukturno slozhnykh tekhnicheskikh kompleksov s pomoshch'yu modeley bayyesovskikh setey doveriya v srede GeNIe. Intellectual Technologies on Transport, № 3, pp. 36-45 (2015)
13. Zhang, R., Brennan, T.J. and Lo, A.W.: The origin of risk aversion. In: PNAS Proceedings of the National Academy of Sciences of the United States of America, Vol. 111, No.50, pp. 17-82 (2014)

14. Shevtsov, U.D., Dudnik, L.N., Aref'yeva, S.A., Fadeyev, Ye.D.: Prognozirovaniye parametrov tekhnicheskogo sostoyaniya dvigateley energeticheskikh ustanovok. In: Politematicheskiy setevoy elektronnyy nauchnyy zhurnal Kubanskogo gosudarstvennogo agrarnogo universiteta, № 132, pp. 508-517 (2017)
15. Kurdyumov S.P., Malinetskiy G.G.: Sinergetika i sistemnyy sintez. Novoye v sinergetike: vzglyad v tret'ye tysyacheletiye, M.: Nauka (2002)
16. Mosleh, A., Groen, F., Hu, Y., Nejad, H., Zhu, D., and Piers, T. Simulation Based Probabilistic Risk Analysis Report. Center for Risk and Reliability, University of Maryland (2004)
17. Vychuzhanin, V., Rudnichenko, N., Shibaeva, N., Konovalov, S.: Devising a method for the estimation and prediction of technical condition of ship complex systems. In: Eastern-European Journal of Enterprise Technologies, Vol. 84, № 6/9, pp. 4 -11 (2016)
18. Vichuzhanin, V., Rudnichenko, N.: Development of the neural network model for prediction failure risk's of complex technical systems components. In: Informatics and mathematical methods in simulation, Vol. 6, №4, pp. 333-338 (2016)
19. Pohorletskyi, D., Gritsuk, I., Zenkin, E., Bulgakov, N., Golovan, A., Vichuzhanin, V.: The Complex Application of Monitoring and Express Diagnosing for Searching Failures on Common Rail System Units. In: SAE International, 2018-01-1773, pp. 1-10 (2018). doi:10.4271/2018-01-1773
20. Vichuzhanin, V., Rudnichenko, N., Mateichyk V., Polyvianchuk A.: Complex Technical System Condition Diagnostics and Prediction Computerization. In: CMIS-2020 Computer Modeling and Intelligent Systems, Ceur-ws.org. 2608 (2020)
21. Kasti, D.: Bol'shiye sistemy. Svyaznost', slozhnost' i katastrofy, M.: Mir (1982)
22. Kochkarov, A.M., Kochkarova, A.N., Khapayeva, L.K.: Modelirovaniye fraktal'nogo razvitiya struktur prosteyshimi predfraktal'nymi grafami. In: Izvestiya YUFU. Tekhnicheskiye nauki, pp. 90-101 (2016)