*Adv Syst Sci Appl 2020; **01**; 83-90*
Published online at http://ijassa.ipu.ru/index.php/ijassa/article/view/825

Network Redundant DCS Configuration Management

N.A. Zakharov[1], V.I. Klepikov[1], D.S. Podkhvatilin[1]

[1)] *"Dozor" subdivision JSC "KEMZ" Moscow, Russia*
E-mail: nazakharov@npp-dozor.ru

Abstract: An approach to the configuration management of a distributed control system with redundant components is proposed. The approach is based on a formalized vector-matrix representation of sets of system components, possible system configurations and functional relationships between them.

Keywords: distributed control system, configuration, configuration management, configuration supervisor, redundancy, fault detection

1. INTRODUCTION

The evolution of distributed control systems (DCS) at the present stage leads to the creation of models of complex automation of processes and production with automatic reconfiguration of their structure without the participation of personnel in order to continue normal operation in the event of failure in their components. DCS with digital communication channels can be represented as a set of jointly and purposefully functioning distributed dynamic objects and, in accordance with the modern theory of systems, is classified as a complex dynamic system.

Currently, a promising concept for the construction of DCS based is on maintenance-free modular electronics [1, 2]. Such a DCS consists of redundant equipment and redundant communication network. This concept implies that the DCS possesses comprehensive system of collection and generalization information about equipment functioning. It uses high-performance algorithms for detection and isolation observable and unobservable failures. The key feature of the DCS under consideration is its capability of deep system reconfiguration based on component redundancy of functional systems and communication network.

2. REDUNDANCY

The most promising way to parry failures in DCS is to use the redundancy of system resources. In general, there are the following types of redundancy:

- system - redundancy of individual components and subsystems;
- structural (architectural) – adaptive reconfiguration of the system structure;
- information - addition to the main signal information on which it is possible to check its reliability (the checksum in the communication channel, estimates on model in dynamic control systems);
- communication - use of duplicated and different types of communication channels;
- algorithmic - using different algorithms to complete the same tasks;
- software – use of different software tools;
- technological - use of a variety of software, information and other technologies;

- time - re-execution of operations with subsequent processing of results;
- semantic - use of redundant semantic structures, digital images of controlled parameters;
- organizational - use of additional systems or subsystems that duplicate functions or information flows of the main system.

3. SCHEDULE-TRIGGERED PROTOCOL

A promising solution to build a communication network is the use of the Schedule-Triggered Protocol (STP) [3, 4]. The network distributed architecture of the control system based on the STP provides a high level of hardware, software, time, communication and information redundancy. This redundancy can be used not only to improve the reliability and fault tolerance of the control system, but also to improve the accuracy and quality of regulation and control. The large bandwidth of the STP channel allows to organize a single information space for all nodes of the distributed system. It allows several nodes to simultaneously perform calculations of control algorithms and transmit the results to the schemes of majority voting or data aggregation.

4. CONFIGURATION SUPERVISOR METHOD

To control DCS redundancy a modified method for configuration supervisor (CS) is offered. In the initial method [5], CS refers to software and hardware modules used to monitor the operability of its configuration, participate in inter-service arbitration to activate its configuration either in case of winning the arbitration, or for parallel work together with other configurations on a common actuator. In the proposed modified CS method, the determination of the integrity of the configuration based on the information about the operability of its components is supplemented with estimates of the integrity of the configurations, formed on the basis of the analysis of the configurations output data. The obtained estimates of the integrity of the configurations are used as feedback to obtain estimates of the operability of the components forming the configuration.

Let's consider DCS based on STP. The DCS has an excessive number of hardware, software, communication and software components (resources). Resources in the general are:

- sensors and input devices;
- actuators and signal output devices;
- computing nodes (controllers);
- communication lines between the nodes and interconnecting systems;
- built-in mathematical models.

In this system, n variants of the sets of the available m components can be organized to perform the specified system functions. Each such variant of the set will be called a configuration.

Different configurations can implement either different or the same system functions, configurations can have completely disjoint component sets, or they can share resources. The designed approach to management of redundancy of the distributed system is providing:

- formalized representation in vector-matrix form of all components, used configurations and sets of components involved in each configuration;
- formalized method for working configurations vector $\boldsymbol{y}$ direct calculation based on the original vector $\boldsymbol{x}$ component of operability;
- calculation of the vector $\hat{\boldsymbol{y}}$ configurations integrities estimates based on configurations operation results;
- formalized the procedure of reverse computations of the vector of estimated components operability based on vector serviceable configurations estimates $\hat{\boldsymbol{y}}$;

- replacement of existing faulty configurations by new or replacement in existing configuration of the failed components by the same serviceable.

5. GAS TURBINE UNIT DISTRIBUTED CONTROL SYSTEM

Let's consider operation of the proposed method on the example of configuration management of a gas turbine unit drive distributed control system (GTU DCS). The block diagram of the GTU DCS is shown in Fig. 1. The system contains k=46 components c_i, i=1...46, which have the following functionality:

- $c_1...c_5$ – $U_1...U_5$ – GTU DCS controllers;
- c_6, c_{12} – L_1, L_2 – redundant STP-bus;
- $c_7...c_{11}$ – $L_{11}...L_{15}$ – controllers $U_1...U_5$ STP-bus L_1 ports;
- $c_{13}...c_{17}$ – $L_{21}...L_{25}$ – controllers $U_1...U_5$ STP-bus L_2 ports;
- c_{18} – L_3 –controller U_5 with the upper level control system communication channel;
- c_{19}, c_{32} – N_{11}, N_{12} – main and backup speed sensors of the low-pressure compressor (LPC);
- c_{20}, c_{33} – N_{21}, N_{22} – main and backup speed sensors of the high-pressure compressor (HPC);
- c_{21}, c_{34} – N_{31}, N_{32} – main and backup speed sensors of the free power turbine (FPT);
- c_{22}, c_{35} – T_{11}, T_{12} – main and backup air temperature sensors;
- c_{23}, c_{36} – P_{11}, P_{12} – main and backup air pressure gauges;
- c_{24}, c_{37} – T_{41}, T_{42} – main and backup temperature sensors of the gas before FPT;
- c_{25}, c_{38} – P_{21}, P_{22} – main and backup combustion chamber pressure sensors;
- c_{26}, c_{39} – Q_{11}, Q_{12} – fuel control valve position main and backup sensors;
- c_{27}, c_{40} – Q_{21}, Q_{22} – LPC guide vanes position main and backup sensors;
- c_{28}, c_{41} – Q_{31}, Q_{32} – HPC guide vanes main and backup position sensors;
- c_{29}, c_{42} – Z_{11}, Z_{12} – main and backup control signals to the fuel control valve;
- c_{30}, c_{43} – Z_{21}, Z_{22} – main and backup control signals to the LPC guide vanes;
- c_{31}, c_{44} – Z_{31}, Z_{32} – main and backup control signals to the HPC guide vanes;
- c_{45} – T_{13}, P_{13}, Ns_3 – information from the upper level control system: air temperature and pressure, set point to the free power turbine speed controller;
- c_{46} – N_{14}, N_{24}, N_{34}, T_{44}, P_{24} – built-in engine model that produces real-time estimates of speed, gas temperature and pressure in the combustion chamber.

The GTU DCS must provide:

- depending on the current free turbine speed setpoint (ns) and the actual values of inlet temperature (t_1) and air pressure (p_1), by regulating the fuel consumption (q_1) by the control signal (z_1), ensure that the required value of the free turbine speed (n_3) is maintained, while preventing the output of the combustion chamber pressure (p_2) and gas temperature (t$_4$) parameters for permissible values, which are also functions of temperature (t_1) and air pressure (p_1).
- depending on the actual values of the engine rotor speeds (n_1, n_2) and the pressure in the combustion chamber (p_2) adjust by means of control signals (z_2, z_3), the guide vanes position (q_2, q_3), which provides the required margin of gas-dynamic stability of the engine.

The control system operation can be described by three functions: f_1 – fuel consumption control, f_2 – LPC guide vanes position control and f_3 – HPC guide vanes position control:

$$\begin{aligned} z_1 &= f_1(ns, n_1, n_2, n_3, t_1, p_1, p_2, t_4, q_1), \\ z_2 &= f_2(n_1, p_2, q_2), \\ z_3 &= f_3(n_2, p_2, q_3). \end{aligned} \tag{1}$$

Analysis of the GTU DCS block diagram (Fig. 1) indicates that the values of the arguments of the functions f_1, f_2 and f_3 can be obtained from various sources (redundant sensors, model, information links with the upper level system). Calculation of functions values can be performed on one or several processors of various controllers, in-system information exchange can be carried out via one of two or simultaneously via both buses of the STP duplicated channel. Issue of control signals both to the fuel control valve and to the guide vanes drives can be made via any of two or simultaneously on both control channels. During GTU DCS operation failures of individual components may occur, there may be failures in the processes of measurement, calculation, data transmission. Due to the hardware, computing, information and time redundancy of the DCS, the GTU can continue operation using remaining serviceable components.

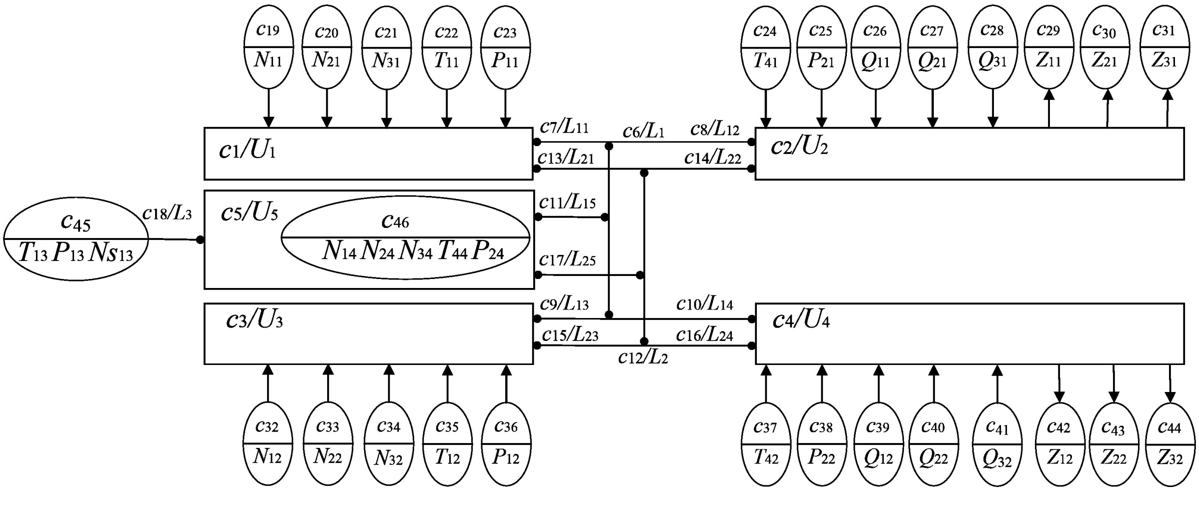

Fig. 1. DCS block diagram

6. CONFIGURATIONS

Let's call configuration a set of components of the system $C = \{c_1, c_2, ..., c_m\}$, providing execution of a certain function. The configuration can provide both the execution of the object control function, i.e. end with the calculation of the output value of the system, and perform intermediate calculations that determine the values of the parameters necessary for other configurations, for example, to calculate the most reliable values of the input parameters of the DCS based on the readings of several sensors. The same function in the control system can be implemented in different configurations depending on which components are currently in good condition.

In complex objects distributed control systems can consist of hundreds of components, which, depending on their current state can be combined into dozens of different configurations to perform certain functional tasks. The formation of such configurations must be performed either in advance at the design stage of the system, or generated automatically in real time, depending on the current situation. Without the use of formal design methods, both approaches are very time-consuming, require a lot of "manual" work at the stages of design and testing of the system. For this reason there is a need to develop analytical methods for representing the sets of configurations of available resources and managing these configurations in real time.

The value of the f_i function formed in some j-th configuration of C_{ij} will be denoted as z_{ij}. In the given system, for the functions f_1, f_2 and f_3 can be formed, for example, three different variants of the calculation of the output values:

$$\begin{gathered}
z^1_1 = f_1(Ns_3, N_{11}, N_{21}, N_{31}, T_{11}, P_{11}, P_{21}, T_{41}, Q_{11}),\\
z^1_2 = f_1(Ns_3, N_{12}, N_{22}, N_{32}, T_{12}, P_{12}, P_{22}, T_{42}, Q_{12}),\\
z^1_3 = f_1(Ns_3, N_{14}, N_{21}, N_{32}, T_{13}, P_{13}, P_{24}, T_{44}, Q_{11}),\\
z^2_4 = f_2(N_{11}, P_{21}, Q_{21}),\\
z^2_5 = f_2(N_{12}, P_{22}, Q_{22}),\\
z^2_6 = f_2(N_{14}, P_{24}, Q_{21}),\\
z^3_7 = f_3(N_{21}, P_{21}, Q_{31}),\\
z^3_8 = f_3(N_{24}, P_{22}, Q_{32}),\\
z^3_9 = f_3(N_{22}, P_{24}, Q_{31}).
\end{gathered} \tag{2}$$

In terms of the functional designation of the component, these variants can be implemented in the following configurations:

$$\begin{gathered}
C^1_1 = \{Ns_3, N_{11}, N_{21}, N_{31}, T_{11}, P_{11}, P_{21}, T_{41}, Q_{11}, U_1, U_2, L_1, L_{11}, L_{12}, L_3, Z_{11}\},\\
C^1_2 = \{Ns_3, N_{12}, N_{22}, N_{32}, T_{12}, P_{12}, P_{22}, T_{42}, Q_{12}, U_3, U_4, L_2, L_{23}, L_{24}, L_3, Z_{12}\},\\
C^1_3 = \{Ns_3, N_{14}, N_{22}, N_{32}, T_{13}, P_{13}, P_{24}, T_{44}, Q_{11}, U_1, U_2, U_5,\\
L_1, L_{15}, L_{12}, L_2, L_{25}, L_{22}, L_3, Z_{11}\},\\
C^2_4 = \{N_{11}, P_{21}, Q_{21}, U_1, U_2, L_1, L_{11}, L_{12}, Z_{21}\},\\
C^2_5 = \{N_{12}, P_{22}, Q_{22}, U_3, U_4, L_{13}, L_{14}, L_2, L_{23}, L_{24}, Z_{22}\},\\
C^2_6 = \{N_{14}, P_{24}, Q_{21}, U_2, U_5, L_1, L_{15}, L_{12}, L_{25}, L_{22}, Z_{21}\},\\
C^3_7 = \{N_{21}, P_{21}, Q_{31}, U_1, U_2, L_1, L_{11}, L_{12}, Z_{31}\},\\
C^3_8 = \{N_{24}, P_{22}, Q_{32}, U_3, U_4, U_5, L_2, L_{23}, L_{24}, Z_{32}\},\\
C^3_9 = \{N_{22}, P_{24}, Q_{31}, U_2, U_3, U_5, L_1, L_{12}, L_{13}, L_{15}, L_2, L_{25}, L_{22}, Z_{31}\}.
\end{gathered} \tag{3}$$

The same configurations in terms of component numbers can be written as:

$$\begin{gathered}
C^1_1 = \{c_{45}, c_{19}, c_{20}, c_{21}, c_{22}, c_{23}, c_{25}, c_{24}, c_{26}, c_1, c_2, c_6, c_7, c_8, c_{18}, c_{31}\},\\
C^1_2 = \{c_{45}, c_{32}, c_{33}, c_{34}, c_{35}, c_{36}, c_{38}, c_{37}, c_{39}, c_3, c_4, c_{12}, c_{15}, c_{16}, c_{18}, c_{42}\},\\
C^1_3 = \{c_{45}, c_{46}, c_{33}, c_{34}, c_{26}, c_1, c_2, c_5, c_7, c_{11}, c_8, c_{12}, c_{17}, c_{14}, c_{18}, c_{31}\},\\
C^2_4 = \{c_{19}, c_{25}, c_{27}, c_1, c_2, c_6, c_7, c_8, c_{30}\},\\
C^2_5 = \{c_{32}, c_{38}, c_{40}, c_3, c_4, c_9, c_{10}, c_{12}, c_{15}, c_{16}, c_{43}\},\\
C^2_6 = \{c_{46}, c_{27}, c_2, c_5, c_6, c_{11}, c_8, c_{17}, c_{14}, c_{30}\},\\
C^3_7 = \{c_{20}, c_{25}, c_{28}, c_1, c_2, c_6, c_7, c_8, c_{31}\},\\
C^3_8 = \{c_{46}, c_{38}, c_{41}, c_3, c_4, c_5, c_{12}, c_{15}, c_{16}, c_{44}\},\\
C^3_9 = \{c_{33}, c_{46}, c_{28}, c_2, c_3, c_5, c_6, c_8, c_9, c_{11}, c_{12}, c_{17}, c_{14}, c_{31}\}.
\end{gathered} \tag{4}$$

Components in formulas (3, 4) can be written in arbitrary order because these formulas are intermediate. They are used to clarify formal representation of configurations. Expressions (4) for sets of c_k components present in C^i_j configurations can be written as a matrix of K dimension ($n \times m$), where n is the number of all configurations considered in the system, m is the number of all components of the system involved in the formation of configurations. The k_{ij} element of the matrix K is 1 if the i-th configuration uses the c_j DCS component (see Fig. 1).

7. STATE AND AVAILABILITY VECTORS

We define a vector $\boldsymbol{x}$ that characterizes the current state of all components of the DCS. If all components of the DCS are operable, the components of the vector $\boldsymbol{x}$ are represented as:

$$x_s = 1, s = 1 \ldots m. \tag{5}$$

Note that all components involved in the configuration are equally involved in the implementation of the system function, regardless of their physical nature and indicators of their own availability.

For the example under consideration, $m = 46$. If there are faulty components in the system, the corresponding components of the vector $\boldsymbol{x}$ are zero. The value of this vector is formed in real time according to the results of the functioning of DCS software and hardware self-diagnostics.

Next, we define the availability vector of configurations $\boldsymbol{y}$, which characterizes the readiness for operation of all considered configurations. $y_i = 1$ if the i-th configuration is healthy, otherwise $y_i = 0$, $i = 1...n$. Configurations availability is a function of the components involved in configurations and their operability.

$$\boldsymbol{y} = F_1(K, \boldsymbol{x}), \tag{6}$$

where F_1 is defined by the expression:

$$F_1 = \bigcap_{s=1}^{m} \left(k_{j,s} \supset x_s\right), \tag{7}$$

the symbol $\bigcap$ denotes the logical "And" function, the symbol $\supset$ denotes the implication function.

The meaning of the expression (7) is that each j-th element of the vector $\boldsymbol{y}$ will be equal to 1, i.e. the j-th configuration will work if only serviceable components are used in this configuration.

In the STP-based DCS, several or all configurations can be executed in parallel or sequentially, which allows you to parry the failures of its components. The number of executable configurations is limited by the computing power of the DCS. If at execution of several configurations different values of the same output parameter are received, it is required to solve two problems: 1) reject incorrect results and mark the configurations which have formed them as faulty; 2) on the basis of the results obtained from the recognized working configurations to form the final agreed output value.

8. DCS COMPONENTS DIAGNOSTICS

An effective way to diagnose the DCS components operability is the inclusion in its composition of the built-in real time model of the object under control [6]. This allows you to get a number of additional features to improve the quality and reliability of control, improve the performance of the system:

- noise and fault filtering;
- the recovery of unmeasured parameters for the diagnosis and management;
- detection of abnormal states of the object and the control system;
- diagnostics of the state and parametric degradation of the object.

Some methods of rejection of incorrect values are considered in [7, 8]. Next, consider the analysis of configurations based on tolerance control.

Let's write down the vector of output values obtained from the results of all n configurations operation.

$$\boldsymbol{z} = [z_1, \ldots z_j, \ldots z_\mathrm{n}]^\mathrm{T} \tag{8}$$

(upper indexes are omitted for clarity) and the initial configuration availability vector

$$\boldsymbol{y} = [y_1, \ldots y_j, \ldots y_\mathrm{n}]^\mathrm{T}. \tag{9}$$

Using the vectors $\boldsymbol{z}^{\min}$ and $\boldsymbol{z}^{\max}$, we set the minimum and maximum output values for each configuration, respectively.

Let's define a tolerance control function as

$$\begin{gathered} F_2(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{z}^{\min}, \boldsymbol{z}^{\max}) = 1, \\ \text{if } (y_j=1) \ \& \ (z^{\min}{}_j \le z_j \le z^{\max}{}_j); \\ \text{else } 0. \end{gathered} \tag{10}$$

The evaluation of the configurations availability vector $\hat{\boldsymbol{y}}$ is defined by the function F_2:

$$\hat{\boldsymbol{y}} = F_2(\boldsymbol{y}, \boldsymbol{z}, \boldsymbol{z}^{\min}, \boldsymbol{z}^{\max}), \tag{11}$$

The zero value of any component of the configuration availability vector indicates that there is one or more failed components of the DCS in the corresponding configuration.

Next, we define the function F_3 as

$$F_3 = \bigcup_{j=1}^{n} \left(k_{j,s} \supset \hat{y}_i\right), \tag{12}$$

the symbol $\bigcup$ denotes a logical OR function; the symbol $\supset$ denotes an implication function. For the vector that characterizes the serviceability evaluation of the DCS components, we can write:

$$\hat{x} = F_3\left(K, \hat{y}\right), \tag{13}$$

The meaning of the expression (13) is that for each s-th element of the state vector, the operability of all $j = 1...n$ configurations, in which it is involved, is checked. If at least one configuration in which this component is involved, i.e. the condition $c_{j,s} \leqslant \hat{y}_j$ is satisfied for at least one $j = 1...n$, then the s-th component is identified as serviceable. If all configurations in which this component is involved are found to be faulty, the component is identified as failed.

9. CONFIGURATION MANAGEMENT ALGORITHM

Configuration management algorithm is the following. The algorithm is executed cyclically. The first step of the algorithm is the formation of the initial vector $\boldsymbol{x}$ of the DCS components serviceability. The vector $\mathbf{v}$ of dimension m formed by the built-in DCS components self-diagnostics means and the vector $\hat{x}$ of components serviceability estimates calculated on the previous cycle are used. At the initial cycle, all components of the vector $\hat{x}$ are assumed to be equal 1. In the second step, based on the original serviceability vector $\boldsymbol{x}$ and the configuration matrix K the vector of configurations availability $\boldsymbol{y}$ is formed. Next, according to the configurations availability verification method (in this example – tolerance control) configurations availability evaluations vector $\hat{\boldsymbol{y}}$ is formed. In the fourth step the DCS components serviceability evaluations vector $\hat{x}$ is calculated. In case of faulty configurations detection, a new configurations matrix K is formed by replacing the used faulty configurations with new ones or replacing the faulty components with serviceable ones in the existing DCS configurations.

10. CONCLUSION

In conclusion it should be noted that both of configurations set and their component composition can be optimized according to various criteria, such as the DCS computing and communication resources load, results interpretation unambiguity. In particular, it is obvious that if each of the system components will be involved in only one specific configuration and absent in all the others, its fault is clearly detected when recognizing this configuration failed. If two or more components are present in only one configuration (or a group of configurations) and are not present in all the others, it is not possible to distinguish their failures due to the failure of this configuration. Optimization of the number and component composition of DCS configurations should be performed taking into account the depth and reliability of the built-in software and hardware self-diagnostics of DCS components.

REFERENCES

1. Jin, H. Lee S., Han S., Jo, H. Kim D. (2012). WiP Abstract: Challenges and Strategies for Exploiting Integrated Modular Avionics on Unmanned Aerial Vehicles. *IEEE/ACM Third International Conference on Cyber-Physical Systems,* Beijing, China, 211-211. https://doi.org/10.1109/ICCPS.2012.34.
2. Wang, L. Sun Y., Guo P., Zhang Y. (2015). An Enhanced Reconfiguration Method for the Second Generation Integrated Modular Avionics, *11th International Conference on Computational Intelligence and Security (CIS),* Shenzhen, China, 433-436. https://doi.org/10.1109/CIS.2015.110.
3. Kopetz, H. (2011) *Real-time systems. Design Principles for Distributed Embedded Applications.* Heidelberg, Germany: Springer. https://doi.org/10.1007/978-1-4419-8237-7.
4. Zakharov, N.A., Klepikov, V.I., Podkhvatilin, D.S. (2013) Sinkhronno-vremennoj protokol dlja raspredelennykh sistem upravlenia [Schedule triggered protocol for distributed control systems], *Avtomatizatsiya v promyshlennosti*, 2, 37-39. [in Russian].
5. Ageev, A.M., Bronnikov, A.M., Bukov, V.N., Gamayunov, I.F. (2017). Supervisory control method for redundant technical systems, Journal of computer and systems sciences international, 56 (3), 410-419. https://doi.org/10.1134/S1064230717030029.
6. Klepikov, V.I., Kalin, S.V., Zakharov, N.A., Podkhvatilin, D.S. (2008). Algoritmicheskoe obespechenie otkazoustojchivosti raspredelennykh system upravlenia [Algorithmic provision of distributed control systems fault tolerance], *Radioelektronni i komp'uterni sistemi*, 34 (7), 43–48. [in Russian].
7. Klepikov, V.I., Podkhvatilin, D.S., Dudorov, Y.N. Sharapov, G.V., Zakharov, N.A. (2011). Information-measuring diagnostics complex for technical maintenance. *Autom Remote Control,* 72 (5), 1089-1094. https://doi.org/10.1134/S0005117911050171.
8. Klepikov, V.I. (2014) *Otkazoustoychivost' raspredelennykh sistem upravleniya* [Fault tolerance of distributed control systems]. Moscow, Russia: Zolotoe sechenie [in Russian].