Mathematical Methods for Detecting and Localizing Failures in Complex Hardware/Software Systems

Olena Syrotkina
Department of Software Engineering
Dnipro University of Technology
Dnipro, Ukraine
ORCID: 0000-0002-4069-6984

Oleksandr Aziukovskyi
Department of Electric Drive
Dnipro University of Technology
Dnipro, Ukraine
ORCID: 0000-0003-1901-4333

Iryna Udovyk
Department of Software Engineering
Dnipro University of Technology
Dnipro, Ukraine
ORCID: 0000-0002-5190-841X

Oleksii Aleksieiev
Department of System Analysis and Control
Dnipro University of Technology
Dnipro, Ukraine
ORCID: 0000-0003-4793-6669

Serhii Prykhodchenko
Department of Software Engineering
Dnipro University of Technology
Dnipro, Ukraine
ORCID: 0000-0002-6562-0601

Leonid Ilyin
Department of Tourism and Hospitality
Lesya Ukrainka Eastern European National University
Lutsk, Ukraine
ORCID: 0000-0002-4180-0544

***Abstract* — This article addresses the problem of creating and applying methods of automatic failure self-diagnostics in complex hardware/software systems (SCADA systems included). We conducted a review of modern methodologies used for solving problems of this class. A structural-logical model was proposed to describe the system's architecture. We developed a mathematical method for diagnosing failures based on analysing changes in the state of system information flows. This included failure diagnostic criteria for the method we developed. As a result, it allowed us to obtain the analytical functional dependencies for the detection and localization of failures in the system's structural modules. Graphs are presented to illustrate the probability of the system's non-failure versus time-of-failure detection and localization. We obtained logical conclusions about improving the quality of system functions using the methods we developed for automatic failure self-diagnostics.**

Keywords — hardware/software systems, SCADA, structural and logical model, failure diagnostics, information flow, failure-free probability.

I. Introduction

During the operation of hardware/software (HW/SW) systems, SCADA (Supervisory Control and Data Acquisition) systems included, abnormal situations may occur caused by failures of some structural modules or the entire system. This leads to a partial or complete loss of control at the operations facility. At the same time, the objects controlled by SCADA systems can be both individual technological aggregates as well as production process facilities. This includes factory departments, industrial enterprises or other facilities such as trunk oil and gas pipelines, high-voltage power supply networks and other critical infrastructure facilities.

In compliance with international standards for the creation and safe operation of SCADA systems [1, 2], failures of main SCADA equipment are grouped into certain classifications. These include failures of critical backbone nodes, field equipment, peripheral equipment, primary converters, data transmission channels and other elements of the multi-level system architecture.

In accordance with the information for failure analysis in automated systems [3], there are certain failures which occur most commonly in peripheral equipment RTU (Remote Terminal Unit). These include backbone nodes and primary converters, as well as data transmission channels from lower-level equipment to the upper levels of SCADA. The equipment used by critical system nodes, such as SCADA servers, database servers, web servers, channel-forming equipment, inter-system and inter-level data exchange equipment, as a rule, have multiple redundancy. These comprise highly reliable power supply systems and, accordingly, are much less prone to failure should unforeseen events occur.

It is known [4] that the reliability of managing any object, except for the reliability of control algorithms and the technical means for their implementation, largely depends on the ability of the enterprise operation to quickly restore the system's operability. In turn, the process of restoring the performance of automated systems can be divided into two stages herein outlined. Stage I – the identification, localization and classification of a failure, and Stage II – the direct elimination of the malfunction and the restoration of the entire system.

It should be noted that the SCADA system elements listed above (in which failures occur most often) are not critical and the operational-dispatching service of the facility management, as a rule, does not record their failure in real time. This is especially true for large and complex multi-level, multi-user and multi-tasking distributed SCADA systems. In this case, the failure of critical nodes of such systems is recorded almost immediately when they occur (for example, failures of database servers, shared access screens, etc.), since such failures occur immediately at many of the operational and non-operational work sites of enterprise personnel.

At the same time, failures of any peripheral nodes of SCADA systems lead to a partial loss of controllability of the object, which is especially unacceptable for objects of critical infrastructure.

Therefore, the urgent task of improving the quality of how SCADA functions is the development of methods for the automatic diagnostics of failures in real time.

II. Analysis of Recent Publications

Modern SCADA systems are distributed complex hardware/software systems operating in real time. Their structure and functionality can change in the process of their life cycle.

There are certain problems in SCADA maintenance which can be attributed to low-level diagnostics as well as a

978-1-7281-6760-2/20/$31.00 ©2020 IEEE

limited ability to restore operability after reversible failures. It can have a negative impact on the control quality for the Technological Control Object (TCO).

Models and methods of failure diagnostics in automated systems are considered in the works of the authors [5 – 11].

Work [5] presents the method of SCADA diagnostics. SCADA and intelligent Distributed Automation (DA) are employed in distributed networks to improve reliability of supply. DA monitoring equipment and control streams of operational data offer a useful insight into underlying circuit conditions. Pole-Mounted Auto Recloser (PMAR) captures current and voltage measurements and isolates distribution of circuit faults. It has the potential to detect the early onset of circuit degradation and monitors its progression. An automated decision support system analyzes PMAR data and corresponding SCADA alarm data for post fault diagnosis and circuit condition assessment.

In paper [6] the authors formulated the attack detection isolation problem. The discrete-time state space model driven by Gaussian noises is used to describe the transient changes detected in stochastic-dynamical systems. The Finite Moving Average (FMA) procedure has been revised for detecting and isolating transient changes in the statistical model.

In paper [7] the authors introduced the Fault Analysis Software Tool (FAST) which is able to perform the structural analysis and fault diagnosis of SCADA. The fault diagnosis algorithms implemented in the tool are described using the well-known two-tank system case study.

In work [8] the authors presented a method of SCADA diagnostics based on a specific, unique alarm (index analysis) generated when a fault occurs. It is generated according to predetermined relevance indexes for a changing set of faults in different contexts. However, this method does not always make it possible to identify the required fault in a changing set. In this case, this fault is considered indistinguishable.

In paper [9] the authors described the application of three-valued logic for software diagnostics. In this case, three-valued logic is represented by one true value and two false values. These are false negative and false positive values. The work shows that in many cases the state of the technical system is faulty if the value of the system parameter is greater or less than the established value. It should be noted that the diagnosis generated by the system during dynamic modeling using the proposed three-valued logic is more accurate.

In work [10] the authors considered the concept of automated system diagnostics based on Latent Variable (LV) models. Defined process parameters were collected by SCADA and exceed the monitoring statistics. These parameters are related to the operation of the equipment and the behavior of the process in the system. Decision support systems based on knowledge can be applied to automate the process of forming an expert judgment. In this case, LV models must be unique and identifiable. Restrictions on the use of LV models consist of limiting the space of latent variables determined by these archival data.

III. Aim of Research

The aim of the research is to improve the functional quality of complex hardware/software systems as demonstrated in the example set out here using a distributed SCADA system. Our example achieves this by providing automatic SCADA diagnostics in real time based on the development of a mathematical method for the detection and localization of failures using the structural and logical model for failure diagnostics (SLMFD).

IV. Problem Statement in General Terms

An example of SCADA structure fragment in general form is described here. This fragment will be used for diagnosing system failures when data flows go through structural elements and various levels of SCADA hierarchy.

For the moment of time t, the set of controlled parameters of the technological control object (CP TCO) is as follows:

$$X(t) = \{x_1(t), x_2(t), \ldots, x_{i_C}(t), \ldots x_{|X|}(t)\}, \qquad (1)$$

This set is measured using primary transducers (PT) and is recorded in specialized controllers – Remote Terminal Units (RTU). Reading $x_{i_C}(t)$ from PT_μ that is connected to RTU_j via data channel $Ch1_\mu$ is provided by a software process running on RTU_j. Server S_γ is connected to RTU_j through wide area network (WAN) using data channels $Ch2_j$. Software processes running on the RTU and server are responsible for data transmission between RTU_j and server S_γ through channel $Ch2_j$. SCADA operating personnel works with server S_γ through the interface of a number of specialized workstations (WKS_η). Server S_γ is connected to WKS_η through wide area network (WAN) using data transmission channels $Ch3_\eta$. Operational and archive information in the form of mnemonic diagrams, reports, graphs, etc. is displayed on WKS_η. Service requests WKS_η are carried out by software processes running on server S_γ.

It is necessary to create the SLMFD, on the basis of which we can develop a mathematical method for detecting and localizing failures in real time by changing the state of the CP TCO when data flows go through structural elements and levels of SCADA hierarchy.

V. Materials and Methods

It is necessary to create the SLMFD, the basis of which we can develop into a mathematical method for detecting and localizing failures in real time. This is accomplished by changing the state of the CP TCO when data flows pass through structural elements and levels of SCADA hierarchy.

The following designations are accepted:

- L – system hierarchy levels corresponding to the levels of possible location of the failure;
- L_1 – system hierarchy levels corresponding to backbone nodes;
- L_2 – system hierarchy levels corresponding to data transmission channels.

In the example analysed, the following levels of SCADA hierarchy are shown in increasing order: PT, $Ch1$, RTU, $Ch2$, S. In this case, the hierarchy level of primary transducers l_{PT}= 1, and the hierarchy level of server l_S=5.

Therefore, the hierarchy levels of the SCADA structure we are examining are described by system (2):

$$\begin{cases} L = \{1,2,3,4,5\}, \\ L_1 \subset L, \quad L_1 = \{1.3.5\}, \\ L_2 \subset L, \quad L_2 = \{2,4\} \end{cases} \tag{2}$$

We define the possible states of the CP TCO for various levels of SCADA hierarchy.

A. Backbone nodes

1) *The value of the controlled parameter is registered at this hierarchy level as reliable (R – reliable).*

2) *The value of the controlled parameter is registered at this hierarchy level as unreliable (UR – unreliable), since, for example:*

a) it is impossible by physical meaning;

b) it is impossible due to the specifics of the controlled process;

c) it is impossible for the current operational mode of the TCO;

d) it is impossible with the current combination of values of other parameters;

e) there are errors detected in transmission and reception protocol;

3) *The parameter value is missing (M – missing).*

B. Data transmission channels

1) *Receiving and transmitting the controlled parameter is reliable (R): the process of receiving and transmitting data has been completed correctly and is considered reliable;*

2) *Receiving and transmitting the controlled parameter is unreliable (UR) since, for example:*

a) transmitting and receiving the controlled parameter is complete, but errors in the data transfer protocol are detected;

b) there is a problem with checksum in the data packet transmitted;

c) incorrect settings of the data transmission channel, etc.

3) *No reception/transmission of the controlled parameter (M): the parameter has not been received or transmitted or this process has not been completed during a custom timeout in order to wait for the reception-transmission process to be completed.*

Several modes of failure detection criteria have been established, for example:

1) *At the system hierarchy levels* $l \in L_1$*, the controlled parameter value is unreliable / missing;*

2) *At the system hierarchy levels* $l \in L_2$*, the reception/transmission of the controlled parameter is unreliable/missing;*

3) *When moving to a higher level of hierarchy* $l \in L_1$*, the value of the controlled parameter changes from reliable to unreliable/missing.*

We define a three-digit set $E_3 = \{0, 1. 2\}$ which corresponds to the controlled parameter states $\{M, UR, R\}$ when $l \in L_1$. We can also accept states $\{M, UR, R\}$ as types of process completion when receiving and transmitting a parameter ($l \in L_2$).

In general terms, the permissible change in the state of a controlled parameter when moving to a higher hierarchy level is described by function $f(x, y, z)$ (see formula 3):

$$\begin{aligned} f(x,y,z) = (2 - 2z^2 - xyz^2 + 2x^2y^2 + x^2y^2z + \\ + x^2yz^2 + xy^2z^2 - 2x^2y^2z^2)(\operatorname{mod} 3), \end{aligned} \tag{3}$$

where x – state of the controlled parameter at the transmitting hierarchy level $l \in L_l$; y – type of process completion when receiving and transmitting the parameter at hierarchy level $l \in L_2$; z – state of the controlled parameter at the receiving hierarchy level $l \in L_l$; $f(x, y, z) = 2$ – the necessary change in the state of the controlled parameter at the receiving hierarchy level, i.e. $(x \vee y) \,\|\!\!\rightarrow \Box z$ (events x or y are the reason why event z is necessary [11]); $f(x, y, z) = 1$ – permissible (possible) change in the state of the controlled parameter at the receiving hierarchy level, i.e. $\neg\,(x \& y) \,\|\!\!\rightarrow \Diamond z$ (events x and y are not the reason for event z, but event z is possible in the system [11]); $f(x, y, z) = 0$ – unacceptable (impossible) change in the state of the parameter at the receiving level of the hierarchy, i.e. $(x \vee y) \,\|\!\!\rightarrow \neg\Diamond z$ (events x or y are the reason why event z is impossible [11]).

Therefore, $f(x, y, z) = 0$ is also a diagnostic feature of failure detection in the system.

We accept the following designations:

- t_0 – time point for measuring controlled parameters in the system using PT;
- Δ_1 – maximum timeout for waiting for the registration of controlled parameters at RTU;
- t_1 – time at the end of registration of controlled parameters at RTU ($t_1 \leq t_0 + \Delta_1$);
- Δ_2 – maximum timeout waiting for the registration of controlled parameters on the server;
- t_2 – the time point for the registration of controlled parameters on the server ($t_2 \leq t_1 + \Delta_2$, $t_2 \leq t_0 + \Delta_1 + \Delta_2$).

At the point of time t_2, based on three-digit set E_3 we can form a diagnostic matrix $D(t_0)$. This matrix must determine a set of states for CP TCO $X(t_0)$ which is distributed through SCADA hierarchy levels. We describe matrix $D(t)$ using system (4).

$$\begin{cases} D(t) = [d_{i_L, i_C}(t)] \\ d_{i_L, i_C}(t) \in E_3 \\ i_L \in L, \quad 1 \leq i_L \leq |L| \\ 1 \leq i_C \leq |X| \end{cases}, \tag{4}$$

where i_L – row index of matrix $D(t)$ corresponding to system hierarchy level according to parameter L in the model; i_C – column index of matrix $D(t)$ corresponding to the index of the controlled parameter in set $X(t)$ of CP TCO; $|L|$ – number of rows in matrix $D(t)$ corresponding to the number of system hierarchy levels; $|X|$ – number of columns in matrix $D(t)$, corresponding to the number of controlled parameters in the system.

We can define the distribution of the CP TCO through *PT*s and data transmission channels between *PT*s and RTUs using a non-decreasing sequence of positive integers K_X:

$$K_X = k_1, k_2, \ldots, k_\mu, \ldots, k_{m_N}, \qquad (5)$$

where μ – ordinal number of sequence member K_X, corresponding to the ordinal number of PT_μ /$Ch1_\mu$; N – the number of RTUs; m_N – the number of *PP*/*Ch*1; $(k_\mu - k_{\mu-1})$ – the number of CP TCO configured to transmit data from PT_μ to RTU_j through $Ch1_\mu$.

We can define the distribution of the CP TCO through RTUs and data transmission channels from RTUs to the server using a non-decreasing sequence of positive integers I_X:

$$I_x = i_1, i_2, \ldots, i_j, \ldots, i_N, \qquad (6)$$

where j – ordinal number of sequence member I_X, corresponding to the ordinal number of RTU_j/$Ch2_j$; N – the number of RTUs; i_N – the number of CP TCO; $(i_j - i_{j-1})$ – the number of CP TCO connected to RTU$_j$/transmitted through $Ch2_j$.

Non-decreasing sequence of positive integers M_K determines the distribution of *PT*/Ch1 through RTUs.

$$M_k = m_1, m_2, \ldots, m_j, \ldots, m_N, \qquad (7)$$

where j – ordinal number of sequence member M_K, corresponding to the ordinal number of RTU; N – the number of RTUs; m_N – the number of PT /Ch1; $(m_j - m_{j-1})$ – the number of *Ch*1 connected to RTU_j.

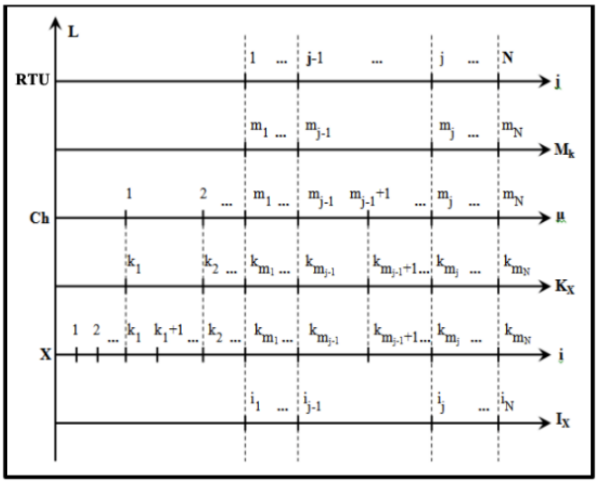
L
RTU
1 ... j-1 ... j ... N
j
m1 ... mj-1 mj ... mN
Mk
Ch
1 2 ... m1 ... mj-1 mj-1+1 ... mj ... mN
µ
k1 k2 ... km1 ... kmj-1 kmj-1+1... kmj ... kmN
KX
X
1 2 ... k1 k1+1 ... k2 ... km1 ... kmj-1 kmj-1+1... kmj ... kmN
i
i1 ... ij-1 ij ... iN
IX

Fig. 1. SCADA structural module configuration

When analysing the diagram (see Fig. 1) it is necessary to fulfil all the conditions of system (8):

From system (8) we can see that the pair of indexes (i_L, i_C) for element d_{i_L,i_C} of diagnostic matrix $D(t)$ is strictly bound to the SCADA structural modules.

$$\begin{cases} i_C \in [1, \ |X|], \\ j \in [1, N], \\ \mu \in [1, m_N], \\ i_j \in I_X, \\ m_j \in M_k, \\ k_{m_j} \in K_X, \\ i_j = k_{m_j} \end{cases} \qquad (8)$$

In order to analyse diagnostic matrix $D(t)$ we apply the elementary function of k-valued logics φ_e (k=3 in our example). This is a characteristic function of the first kind for value $e \in E_k$:

$$\varphi_e(x) = \begin{cases} 1, & x = e, \quad e \in E_k \\ 0, & x \neq e, \quad e \in E_k \\ E_k & = \{0, 1, k-1\} \end{cases} \qquad (9)$$

In general, the function of detecting and localising a failure in the system g_2 is based on the analysis of diagnostic matrix $D(t)$, as follows:

$$g_2(i_L, \alpha, \beta, t) = \neg\left(\mathop{\&}_{i_C=\alpha}^{\beta} \phi_2(d_{i_L, i_C}(t)) \right). \qquad (10)$$

Description of correspondence of argument combinations such as i_L, α and β for function g_2 (the function of detection and localization of failures in the system) is shown in [14].

When analysing diagnostic matrix $D(t)$ we accept that all the functional components of the distributed SCADA system involved in the transmission of the CP TCO from PTs to the server are considered faultless at the time of point t. This is true if the following condition (11) is fulfilled for row i_L = |L| of matrix $D(t)$ corresponding to the server's hierarchy level:

$$\mathop{\&}_{i_C=1}^{|X|} \phi_2(d_{/L/, i_C}(t)) = 1. \qquad (11)$$

The method described in this paper for automatic detection and localization of failures is based on the reliability analysis of the CP TCO when data flows go through structural modules and levels of system hierarchy. It can be applicable to any information and control system of any topology.

We derived the following analytical dependencies:

- Determination of the allowable change in the state of the controlled parameter when moving to an upper hierarchy level;
- detection and localization of failures in the system.

This allows automatic detection and localization of failures in complex HW/SW systems to be performed in real time.

Additional information about the fault diagnostics method in SCADA (based on the model of changing the state of information flows) is given in [12−14].

VI. Results and Discussion

Fig. 2 shows schematically the periods of operability restoration for an arbitrary structural module (or the group of structural modules) of SCADA.

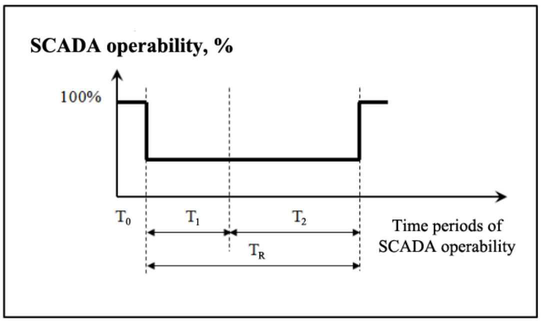

Fig. 2. Periods of SCADA operability

The following designations are presented in Fig. 2:

- T_0 – operating time to failure;
- T_R – operability restoration time;
- T_1 – fault detection and localization time;
- T_2 – fault correction time.

As can be seen from Fig. 2, the total recovery time T_R of SCADA performance is largely dependent on time period T_1. In many cases, this time may even exceed time T_2 needed to repair the failure.

In order to analyse the influence of time period T_1 on the reliability of the system's structural module, we accept exponential distribution as a function of distribution of the operating time to failure T_0.

It is known that each structural module of a system operates under a certain load and has a maximum load at which it remains operational. At peak loads, and due to unforeseen external influences, a sudden failure is possible [4].

It is also known that in the theory of random processes there is evidence that "under certain conditions, the time until the first random process crosses a certain threshold level has an exponential distribution" [4].

The formula for calculating the average group failure rate is as follows [4]:

$$\lambda_i = \frac{1}{\left(m_i * \overline{T}_{0i}\right)} < -\ln p_{0i}(t), \qquad (12)$$

where λ_i – average group failure rate for calculating the reliability indicators of a certain average element of the group. At the same time, assumptions are made that all elements of the group are equally reliable and their failure rates are constant; m_i – the number of elements in the i^{th} group; $\overline{T}_{0i}$ – the average time to failure for the i^{th} group of the system's structural elements; p_{0i} – the required value of the system's reliability function for the i^{th} group of the system's structural elements.

Fig. 3 represents the system's reliability function for an exponential distribution of time to failure depending on the average group failure rates.

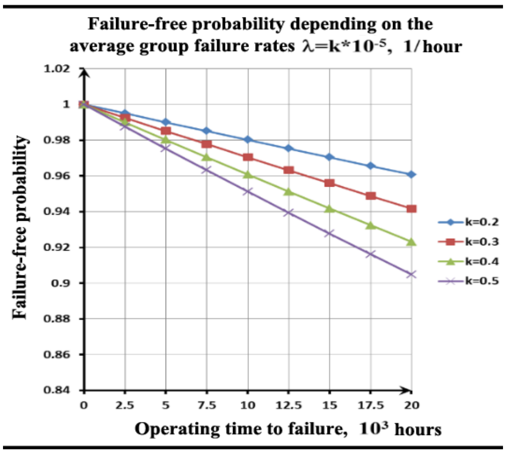

Fig. 3. Graphs of the system's reliability function for an exponential distribution of time to failure

We accept the Markov model as a functional basis for the structural group of system elements. This model is characterized by the following [4]:

- the future state of the elements depends only on the current state and does not depend on its background;
- all time intervals T_0 (operating time to failure), T_1 (time to detect a latent failure) and T_2 (fault correction time) (see. Fig. 2) have the property of independence and have an exponential distribution.

For Markov models, failure-free application probability $P_{pr}(t)$ (i.e. the probability that latent failure would not occur until operating time to failure occurs – not at the initial moment of time) can be defined as follows [4]:

$$P_{pr}(t) = \frac{1}{(1+\beta * \overline{T}_{\det})} e^{-\lambda t}, \qquad (13)$$

where β – latent failure component; $\overline{T}_{\det}$ – average latent failure detection time.

Dependence of reducing failure-free probability $P(t)$ on parameters β and $\overline{T}_{\det}$ is shown in Fig. 4.

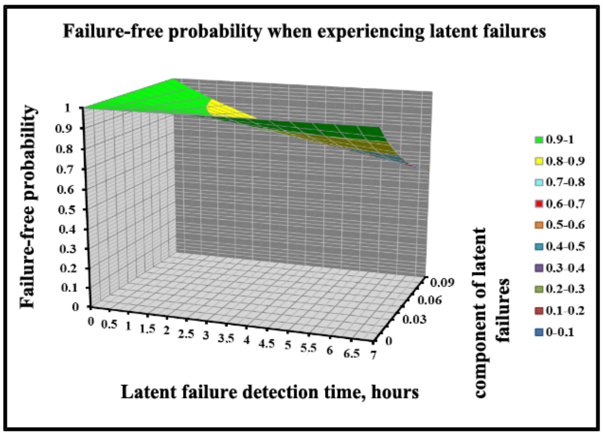

Fig. 4. Dynamics of reducing failure-free probability when experiencing latent failures

Dependence of the difference between the probabilities of failure-free operation on the parameters β and $\overline{T}_{\text{det}}$ with and without the presence of latent failures is shown in Fig. 5.

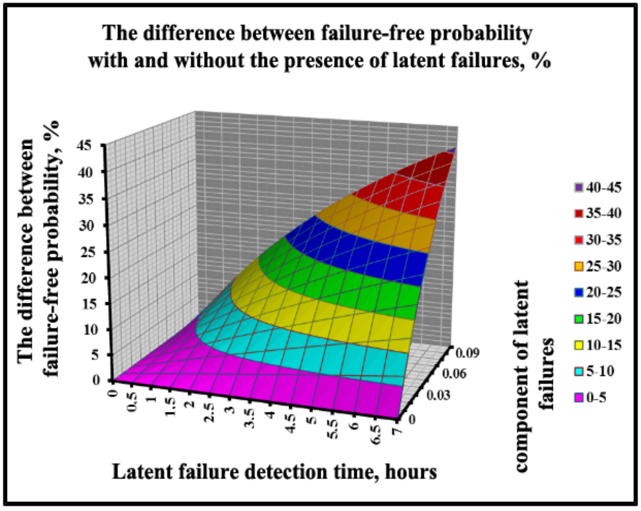

Fig. 5. Dynamics of increasing the difference between failure-free probability without and with the presence of latent failures

Fig. 4 and 5 show the values of failure-free probability (FFB) calculated for the average group failure rate $\lambda = 0{,}2{*}10^{-5}$, 1/*hour*.

The analysis results of how a latent failure influences failure free probability are shown in Table 1.

Therefore, minimizing the latent failure detection period T_1 to $\overline{T}_{\text{det}} = 0.1\ h = 6\ min$ allows us to prevent reduction of FFB in the event of latent failures by more than 1% (see Fig. 5, Table 1). This increases FFB by 7%...8% in comparison with the average data of the intervals considered for parameter changes ($\overline{T}_{\text{det}} = 3h$, β=3%).

TABLE I. DEPENDENCE OF REDUCING FAILURE-FREE PROBABILITY ON FAILURE DETECTION TIME AND COMPONENT OF LATENT FAILURES

№	λ, 1/hour	β, %	$\overline{T}_{\text{det}}$, hour	Difference of FFB, %
1	***2***	***3***	***4***	***5***
1	$0.2{*}10^{-5}$	1	0.1	0.1
2		3	0.1	0.3
3		5	0.1	0.5
4		1	3	2.9
5		3	3	8.25
6		5	3	13.03
7		1	7	6.54
8		3	7	17.33
9		5	7	25.9
10	$0.5{*}10^{-5}$	1	0.1	0.09
11		3	0.1	0.27
12		5	0.1	0.45
13		1	3	2.62
14		3	3	7.43
15		5	3	11.74
16		1	7	5.89
17		3	7	15.62
18		5	7	23.33

VII. CONCLUSIONS

The method of failure detection and localization considered in this article for complex hardware/software systems is based on the analysis of changes in the state of information flows. It allows automatic detection and localization of failures in the system to be performed during its operation in real time. This method is applicable to hardware/software systems of any topology.

This approach allows us to reduce the system's recovery time to nearly the time it would take to recover from that failure.

Minimizing the detection period of latent failures increases the system's failure-free probability by an average of 7% to 8%.

The use of this method for automatic failure diagnostics is a good prerequisite for creating methods of automatic self-recovery of distributed information and control systems after reversible failures in real time.

REFERENCES

[1] K. Stouffer, J. Falco, and K. Kent, "Guide to supervisory control and data acquisition (SCADA) and industrial control systems security," Recommendations of the National Institute of Standards and Technology: NIST Special Publication 800-82, pp. 1–170, April 2013.

[2] IEC 61508 Overview Report. "A Summary of the IEC 61508 Standard for Functional Safety of Electrical/Electronic/Programmable Electronic Safety-Related Systems," Exida, Sellersville, USA, ver. 2.0, pp. 6–29, January 2006.

[3] Positive Technologies Research Blog Review, "ICS/SCADA Safety in Numbers," Positive Technologies, pp. 1–29, October 2013.

[4] J. Cano and D. Rios, "Reliability forecasting in complex hardware/ software systems," First International Conference on Availability, Reliability and Security (ARES'06), pp. 1–5, September 2006.

[5] X. Wang, S. Mcarthur, S. Strachan, and B. Paisley, "Decision support for distribution automation: data analytics for automated fault diagnosis and prognosis," 24th International Conference & Exhibition on Electricity Distribution (CIRED), Iss. 1, pp. 1022–1026, June 2017.

[6] V. Do, "Statistical detection and isolation of cyber-physical attacks on SCADA systems,' 43rd Annual Conference of the IEEE Industrial Electronics Society (IECON), pp. 3524–3529, 2017.

[7] J. Duatis, C. Angulo, and V. Puig, "FAST: A fault analysis software tool," 2014 IEEE Conference on Control Applications (CCA), Antibes, France, pp. 376–381, October 2014.

[8] S. Charbonnier, N. Bouchair, and P. Gayet, "Analysis of fault diagnosability from SCADA alarms signatures using relevance indices," 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), San Diego, CA, USA, pp. 2739–2744, October 2014.

[9] A. Ligeza, "A 3 – Valued Logic for Diagnostic Applications," Papers of workshop "Diagnostic REAsoning: Model Analysis and Performance DREAMAP 2012", Montpellier, France, pp. 1–4, 2012.

[10] J. MacGregor and A. Cinar, "Monitoring, fault diagnosis, fault-tolerant control and optimization: Data driven methods," Computers & Chemical Engineering, vol. 47, pp. 111–120, 2012.

[11] J. Garson, "Modal Logic for Philosophers," 2nd Edition, Cambridge University Press, 506 p., January 2014.

[12] M. Alekseyev, I. Udovyk, and O. Syrotkina, "Application of the predicate system in the structural and logical model of SCADA failure diagnostics," Systems and Means of Artificial Intelligence: Nauka i Osvita, Kyiv, pp. 14–19, 2017.

[13] O. Syrotkina and M. Alekseyev "Software diagnostics for reliability of SCADA structural elements," Power Engineering and Information Technologies in Technical Objects Controls: Taylor & Francis Group, London, pp. 259–265, 2016.

[14] O.Syrotkina, "SCADA analytical model of fault detection and localisation," Dnipropetrovsk: Metallurgical and Mining Industry, Iss. 5, pp. 112–115, 2014.