EVENT MONITORING OF PARALLEL COMPUTATIONS

ALEXANDER M. GRUZLIKOV [a], NIKOLAI V. KOLESOV [a,*], MARINA V. TOLMACHEVA [a]

[a] Concern CSRI Elektropribor
ITMO University, 49 Kronverksky Pr., St. Petersburg, Russia
e-mail: `kolesovnv@mail.ru`

The paper considers the monitoring of parallel computations for detection of abnormal events. It is assumed that computations are organized according to an event model, and monitoring is based on specific test sequences.

Keywords: parallel computations, monitoring, discrete event system, real time systems.

1. Introduction

The quality of computing system (CS) diagnosis critically affects the system performance. Therefore, an improvement of diagnostic tools remains a pressing problem which has various aspects (Patton *et al.*, 2000; Zhirabok, 2009; Chanthery and Pencole, 2009). Possible solutions rely on a set of tools, among which the monitoring of computations is of considerable importance. In practice, in most cases we have to deal with the monitoring of distributed, i.e., parallel computations. The aim of the monitoring considered in this paper is to detect faults in data exchanges between the program modules (PMs) of which the CS software consists. Faults can be caused not only by hardware failures, but they may also result from design errors in the organization of computations and the software used.

Assume that the organization of computations in a computing system is based on an asynchronous approach (Cheng, 2002). This means that the PMs of the system process the data as soon as they are ready, rather than according to a predefined schedule, that is, asynchronously. If the executive processor is busy, the data are queued up. The asynchronous approach is formalized by a Petri net (Peterson, 1981), initially proposed to describe asynchronous interaction of computation modules. No wonder that it is this model that is widely used in solving the problem of computation monitoring, whose different versions are discussed in numerous publications (Kolesov, 1988; Wu and Hadrjicostis, 2005; Cabasino *et al.*, 2011; Lefebvre,

2012; Chanthery and Pencole, 2009; Zhirabok, 2009). The object of monitoring is a discrete event system, with the system operation represented as a sequence of events.

We also use a model of a discrete event system in which the analyzed events are reception and output of the data from the system PMs. A specific feature of our approach is that the monitoring problem is solved within the test diagnosis concept. The system is made redundant due to the introduction of an event model.

The authors apply the described approach to real-time systems, although it can be extended to other systems as well. A feature of real-time systems is periodicity of input data flows and their processing.

This paper is structured as follows. Section 2 considers the problem statement, Section 3 is concerned with the synthesis of the computation model, and Sections 4 and 5 discuss some relevant results from the theory of diagnosis of periodically time-varying discrete systems. We also give an example to illustrate the problem solution.

2. Problem statement

To explain the principle of the proposed approach, let us consider a structure (an information graph) of a hypothetical system S (Fig. 1). The three functionally related program modules PM_1, PM_2, and PM_3 implemented in the system can be located either in the same or in different processors. Each program module generates output data (y_1 for PM_1, y_2 for PM_2, and y_3 for PM_3) from the input data (u_1 for PM_1, u_2 and y_3 for PM_2, and y_1 and y_2 for PM_3). The input data periods are

---
*Corresponding author

equal. The queues at the PM input are shown as hatched rectangles.

Figure 2 shows the functional structure of event monitoring for the example under consideration. The monitoring system (MS) is represented as a combination of three traditional modules: a test data generator (TDG), a sample response generator (SRG) and a comparator (C). The MS generates tests (data words) for system S and analyzes its response. Real data words are processed by standard algorithms PM_1, PM_2, and PM_3 in each of the redundant PMs (PM'_1, PM'_2, and PM'_3). At the same time, test data words are processed by special supplementary algorithms f_{m1}, f_{m2}, and f_{m3} responding to data reception/output events. The processing results are included in the output data. The test data are transmitted together with the real data. As a result, any fault in the transmission of real data would mean also a fault in the transmission of test data. Thus, diagnosing the test data processing we can detect faults in the procedures of data exchange. The class of faults includes different changes in the sets of the PMs involved in data exchange, in particular, miss and emergence of new exchanges, which may result in rearrangement of PMs in a certain computational path. Formally, the class of faults is defined in terms of the dynamic periodically time-varying model proposed below, namely, the MS detects any fault that causes changes in the matrix of this model. Consequently, a challenge is to select proper test data and an algorithm for their processing in the PM.

Thus, the problem of designing supplementary algorithms is considered below. The issue of testing these algorithms is also discussed in the paper.
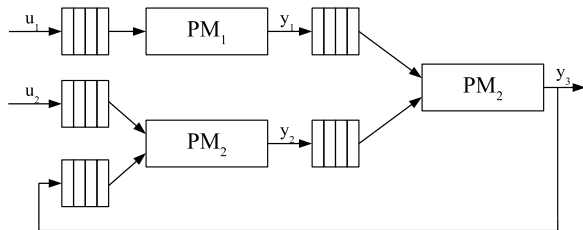

Fig. 1. Structure (an information graph) of a system.

3. Synthesis of a system event model

One of the critical issues of the problem discussed here is proper selection of a test data processing algorithm in each PM, which is not a trivial task. Note, first of all, that the combined test data processing algorithm f_Σ in S is a composition of the algorithms in the PMs. Like the main processing algorithm, it is implemented as a parallel computation process in accordance with the asynchronous approach. In a sense, the test data processing algorithm is an event model of a real data processing algorithm,

and therefore, a system model. Needless to say that the observable result of test data processing is not the events themselves, but the output symbolic sequence which can be used to restore the sequence of analyzed events and to determine whether it is correct or incorrect. From the above it follows that the problem of the synthesis of a test data processing algorithm can be formulated as a problem of the synthesis of a system model.

However, the algorithm f_Σ is inconvenient as a system model to be used for the solution of the monitoring problem. In our opinion, the model should meet the following three requirements. Firstly, it should be simple enough in order not to involve significant computational resources and to allow a synthesis for any graph of intermodule data communication. Secondly, it is desirable that it should be dynamic because the algorithms for designing tests for dynamic systems are known. Thirdly, it should be adequate so that the test designed in accordance with the model could detect all the faults considered in this paper.

Below, we propose a universal approach to the synthesis of f_Σ, suitable for any graph of intermodule information communication. Using special techniques,
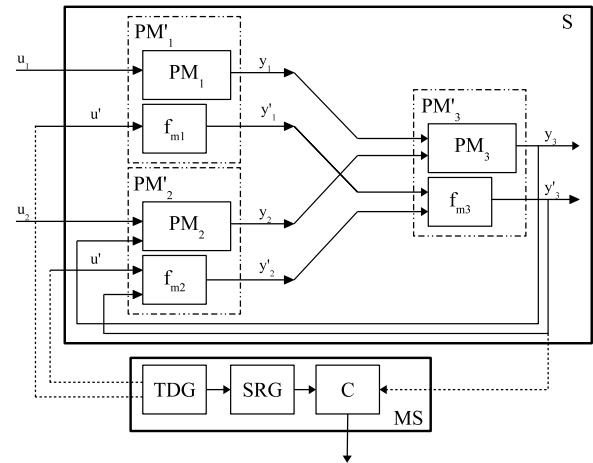

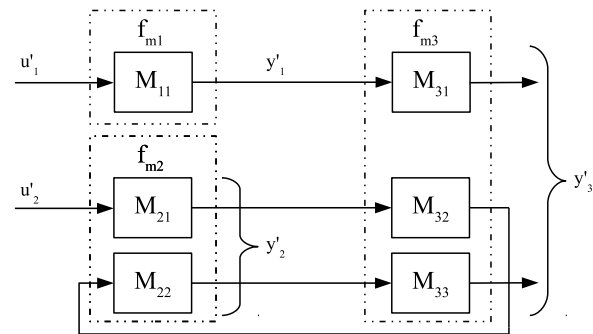Fig. 2. Functional structure of event monitoring.

Fig. 3. System event model.

we reduce the analysis of arbitrary information graphs to the analysis of some standard (primitive) graphs, namely, chains.

To construct the system model, let us use the following procedure. At the first step, analyzing the information graph, we find a set of computation paths covering its edges. By a computation path, we mean a sequence of edges and nodes connecting the input and output of the graph. At the second step of the procedure, a chain of M dynamic elements in the system model is assigned to each path, where M is the number of PMs through which the path goes. If PM_i is included in the path and it is an l-th PM in the chain, element M_{il} is assigned to this PM_i.

For the example considered in Fig. 1, the cover is provided by two paths with edges u_1', y_1', y_3' and $u_2', y_2', y_3', y_2', y_3'$. The primed letters denote the test data transmitted in arrays. The second path includes the cycle in order to cover the information feedback from the system output to the input PM_2. Thus, the system model becomes a set of independent chains.

By independence, we mean that the result of computation in one chain does not depend on the results of computation in the other chains. It can be inferred that the result of model computation is formed as an array of the values of the arguments that were subjected to independent processing.

Since the model of the system consists of M independent chains, it is possible to provide its own algorithm for the monitoring of each chain in the MS. Therefore, the problem of system monitoring is reduced to chain monitoring, which is much easier to solve.

Figure 3 shows the system event model for the example under consideration, where model PM_1 contains one element M_{11}, model PM_2—two elements M_{21} and M_{22}, and model PM_3—three elements M_{31}, M_{32}, and M_{33}. Array y_3', the output of the third system, has a specific structure: it consists of two parts, one formed in elements M_{31} and M_{33} intended for the MS, and the other formed in M_{32} intended for PM_2.

Let us refine the model for the supplementary algorithm. From technical diagnosis theory (Patton *et al.*, 2000; Zhirabok, 2009) it is known that tests for linear objects are rather simple. It is clear that since we deal with the analysis of an event sequence (a sequence of task solutions), this algorithm should be dynamic. As a result, we come to the conclusion that a linear dynamic model for an element is preferred,

$$x_{i,l}(t+1) = f_{i,l}x_{i,l}(t) + g_{i,l}u_{i,l}(t),$$
$$y_{i,l}(t) = h_{i,l}x_{i,l}(t), \quad i = \overline{1,m}, \quad l = \overline{1,m_i}, \quad (1)$$

where $x_{i,l}$, $u_{i,l}$ and $y_{i,l}$ are the state, input and output vectors, respectively, $f_{i,l}$, $g_{i,l}$ and $h_{i,l}$ are the matrices of dynamics, input and output of the l-th element in the i-th

PM model, m is the number of PMs, and m_i is the number of elements in PM_i.

Each element of the PM model processes data only from one array using the same algorithm. Unlike an element, a PM generally processes several arrays; therefore, its model (additional algorithm) contains a parallel composition of several elements.

The transmitted arrays may have different dimensions depending on the number and structure of the elements forming the PM model. The elements of the PM model trigger at different instants of time and, in a general case, process data by different algorithms. This means that the PM model is time-varying.

In refining the system model, remember that it consists of several chains; thus, by refining the chain model, we refine the system model. For simplicity, it makes sense to reduce the analysis for monitoring to the analysis of a dynamic system, the more so that the monitoring of dynamic systems is well understood. However, describing a chain as a dynamic system cannot be considered quite adequate because in this case the parallel process (the chain working as a pipeline) is described as a sequential one, in which only one operation is possible at each time instant.

Nevertheless, this description can be used to solve the event-monitoring problem. For the monitoring problem it is essential that, under nominal behavior, the sequence and contents of output events generated by the system and calculated on the basis of a sequential dynamic model should coincide, but they should be different at faults considered here. In what follows, we show how to provide this condition.

Now, let us make a temporary assumption that only one data exchange is possible at a certain instant of time. Later on, we will give up this unreal assumption. Consider a chain of L elements, each of them described by Eqn. (1). In view of the above assumption, we represent a chain as a linear dynamic system in which the state vector is the vector $x(t)$ consisting of the state vectors of elements $x_i(t), i = \overline{1,L}$ comprised in the chain.

Now, let us describe the data exchange between two adjacent elements: data reception from the MS to the chain input element and data output from the chain output element to the MS.

Assume that information is transmitted from the PM_i element to the element PM_j. Let us write down the equation of chain for the exchange considered assuming that the number of PMs is L. This process is described by the following equation with the block dynamic matrix (we

omit the zero elements of the dynamic matrix)

$$
x(t+1)
$$

$$
= \begin{bmatrix} x_1(t+1) \\ x_2(t+1) \\ \vdots \\ x_i(t+1) \\ \vdots \\ x_j(t+1) \\ \vdots \\ x_L(t+1) \end{bmatrix}
$$

$$
= \begin{bmatrix} E & & & & & & \\ & E & & & & & \\ & & \ddots & & & & \\ & & & f_i & & & \\ & & & & \ddots & & \\ & & & g_j h_i & & f_j & \\ & & & & & & \ddots \\ & & & & & & & E \end{bmatrix}
$$

$$
\times \begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_i(t) \\ \vdots \\ x_j(t) \\ \vdots \\ x_L(t) \end{bmatrix}. \tag{2}
$$

The transformation changes only the states of the elements of the interacting PMs (PM_i and PM_j). Hence the states of other elements remain unchanged. The fact that the PM state remains unchanged is described by the identity diagonal matrix E in the corresponding block of the block matrix diagonal. It is assumed that both PM_i and PM_j element models trigger in accordance with (1) when information is transmitted from PM_i to PM_j. In this case, PM_j does not receive information; therefore, the second summand in the dynamics equation is zero ($g_i = 0$), i.e.,

$$
x_i(t+1) = f_i x_i(t), \quad y_i(t) = h_i x_i(t).
$$

The second index of variables denoting the element number in the PM is omitted for simplicity.

After the information $u_j = y_i = h_i(x_i)$ is received, there is no data output from PM_j, so the output is zero ($h_j = 0$) and the dynamic equation

$$
\begin{aligned}
x_j(t+1) &= f_j x_j(t) + g_j u_j(t) \\
&= f_j x_j(t) + g_j h_j x_i(t)
\end{aligned}
$$

corresponds to the j-th row of Eqn. (2).

The equation describing data reception from the MS looks simpler because the model of only one receiving PM_j takes part in the exchange and triggers

$$
x(t+1)
$$

$$
= \begin{bmatrix} x_1(t+1) \\ \vdots \\ x_j(t+1) \\ \vdots \\ x_L(t+1) \end{bmatrix}
$$

$$
= \begin{bmatrix} E & & & & \\ & \ddots & & & \\ & & f_j & & \\ & & & \ddots & \\ & & & & E \end{bmatrix} \begin{bmatrix} x_1(t) \\ \vdots \\ x_j(t) \\ \vdots \\ x_L(t) \end{bmatrix}
$$

$$
+ \begin{bmatrix} 0 \\ \vdots \\ g_j \\ \vdots \\ 0 \end{bmatrix} u(t).
$$

Since the data are not output to the MS ($h_j = 0$), the output is zero.

The equation describing data output to the MS looks simpler, too, because the model of only one output PM_s takes part in the exchange and triggers

$$
x(t+1) = \begin{bmatrix} x_1(t+1) \\ \vdots \\ x_s(t+1) \\ \vdots \\ x_L(t+1) \end{bmatrix}
$$

$$
= \begin{bmatrix} E & & & & \\ & \ddots & & & \\ & & f_s & & \\ & & & \ddots & \\ & & & & E \end{bmatrix} \begin{bmatrix} x_1(t) \\ \vdots \\ x_s(t) \\ \vdots \\ x_L(t) \end{bmatrix},
$$

$$
y(t) = \begin{bmatrix} 0 & \cdots & h_s & \cdots & 0 \end{bmatrix} \begin{bmatrix} x_1(t) \\ \vdots \\ x_s(t) \\ \vdots \\ x_L(t) \end{bmatrix}.
$$

Let us write the equations of the chain in general form:

$$
\begin{aligned}
x(t+1) &= \tilde{F}(j)x(t) + \tilde{G}(j)u(t), \\
y(t) &= \tilde{H}(j)x(t), \quad j = \overline{1, L+1}.
\end{aligned} \tag{3}
$$

These equations describe $L - 1$ interelement exchanges and two exchanges with the MS (data reception and output). The matrices in these equations depend on the number of the cycle (exchange), i.e., the model is time-varying. Since we consider real-time systems in which the input data flow and information processing or control are periodic, the above model is a periodically time-varying one with a period of $L + 1$. If the system model has M chains, its description is given by

$$x(t + 1) = \tilde{F}_k(j)x(t) + \tilde{G}_k(j)u(t),$$
$$y(t) = \tilde{H}_k(j)x(t) \quad j = \overline{1, L + 1}, \quad k = \overline{1, M}. \tag{4}$$

Example 1. Consider general matrices of the model (4) of the distributed system for the example in Fig. 1. The description of the system includes two chains, with two elements in the first one and four in the second:

(i) the first chain,

$$\tilde{F}_1(1) = \begin{bmatrix} f_{1,1} & 0 \\ 0 & E \end{bmatrix},$$

$$\tilde{F}_1(2) = \begin{bmatrix} f_{1,1} & 0 \\ g_{1,2}h_{1,1} & f_{1,2} \end{bmatrix}$$

$$\tilde{F}_1(3) = \begin{bmatrix} E & 0 \\ 0 & f_{1,2} \end{bmatrix},$$

$$\tilde{G}_1(1) = \begin{bmatrix} g_{1,1} \\ 0 \end{bmatrix},$$

$$\tilde{G}_1(j) = 0, \quad j = \overline{2, 3},$$

$$\tilde{H}_1(j) = 0, \quad j = \overline{1, 2},$$

$$\tilde{H}_1(3) = \begin{bmatrix} 0 & h_2 \end{bmatrix},$$

(ii) the second chain,

$$\tilde{F}_2(1) = \begin{bmatrix} f_{2,1} & 0 & 0 & 0 \\ 0 & E & 0 & 0 \\ 0 & 0 & E & 0 \\ 0 & 0 & 0 & E \end{bmatrix},$$

$$\tilde{F}_2(2) = \begin{bmatrix} f_{2,1} & 0 & 0 & 0 \\ g_{2,1}h_{2,2} & f_{2,2} & 0 & 0 \\ 0 & 0 & E & 0 \\ 0 & 0 & 0 & E \end{bmatrix},$$

$$\tilde{F}_2(3) = \begin{bmatrix} E & 0 & 0 & 0 \\ 0 & f_{2,2} & 0 & 0 \\ 0 & g_{2,2}h_{2,1} & f_{2,3} & 0 \\ 0 & 0 & 0 & E \end{bmatrix},$$

$$\tilde{F}_2(4) = \begin{bmatrix} E & 0 & 0 & 0 \\ 0 & E & 0 & 0 \\ 0 & 0 & f_{2,3} & 0 \\ 0 & 0 & g_{2,4}h_{2,3} & f_{2,4} \end{bmatrix},$$

$$\tilde{F}_2(5) = \begin{bmatrix} E & 0 & 0 & 0 \\ 0 & E & 0 & 0 \\ 0 & 0 & E & 0 \\ 0 & 0 & 0 & f_{2,4} \end{bmatrix},$$

$$\tilde{G}_2(1) = \begin{bmatrix} g_{2,1} \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$\tilde{G}_2(j) = 0, \quad j = \overline{2, 5},$$

$$\tilde{H}_2(j) = 0, \quad j = \overline{1, 4},$$

$$\tilde{H}_2(5) = \begin{bmatrix} 0 & 0 & 0 & h_{2,4} \end{bmatrix}.$$

Now, we give up the assumption that only one exchange is possible in the system at each instant of time. Let us show that, in this case, the design of system monitoring is nonetheless possible with the periodically time-varying model described above.

To proceed to the proposed dynamic description of the model, all elements of the chain are formally replaced by one element with variable matrices, repeated with a period of data input. Thus, we come to a periodically time-varying description of the chain. This replacement is illustrated in Fig. 4 for the simplest case of a two-element chain. The chain provides for three exchanges: reception to element 1, transmission from element 1 to element 2, output from element 2. Computations in the chain are actually realized in parallel, which is shown in Fig. 4(a) by two time diagrams for two portions of input data separated by period T. Here, all three stages of computations in the chain are of the same duration. The intervals limited by broken lines correspond to different exchanges with the matrices indicating these intervals. In the model applied in the MS, these computations are performed sequentially (Fig. 4(b)). The intervals shown by broken lines correspond to a fixed set of element matrices for a fixed portion of input data. For clarity, we use different time scales in Figs. 4(a) and 4(b).

Let us show a possibility of using this model for designing a system test, namely, the adequacy of this model. Firstly, note that a model composed of independent chains corresponds to both a fault-free system and a faulty system. In the second case, the model of at least one chain will be distorted, as compared with the first case. Any fault considered causes changes in the matrix of this model, because the class of faults includes different changes in the sets of the PMs involved in data exchange. Secondly, as a response to equal input sequences, any chain and its model form equal output sequences that are only different in the time position of the output vectors of these sequences. This difference is due to the fact that, for a chain, the time position depends on how long the basic information is processed in the system PM. This time is usually long and not known in advance,

$$\begin{array}{c|c|c}
\widetilde{F}(1) & \widetilde{F}(2) & \widetilde{F}(3) \\
\widetilde{G}(1) & \widetilde{G}(2) & \widetilde{G}(3) \\
\widetilde{H}(1) & \widetilde{H}(2) & \widetilde{H}(3)
\end{array}$$

Processing of portion 1

$$\begin{array}{c|c|c}
\widetilde{F}(1) & \widetilde{F}(2) & \widetilde{F}(3) \\
\widetilde{G}(1) & \widetilde{G}(2) & \widetilde{G}(3) \\
\widetilde{H}(1) & \widetilde{H}(2) & \widetilde{H}(3)
\end{array}$$

T Processing of portion 2

(a)

$$\begin{array}{c|c|c|c|c|c}
\widetilde{F}(1) & \widetilde{F}(2) & \widetilde{F}(3) & \widetilde{F}(1) & \widetilde{F}(2) & \widetilde{F}(3) \\
\widetilde{G}(1) & \widetilde{G}(2) & \widetilde{G}(3) & \widetilde{G}(1) & \widetilde{G}(2) & \widetilde{G}(3) \\
\widetilde{H}(1) & \widetilde{H}(2) & \widetilde{H}(3) & \widetilde{H}(1) & \widetilde{H}(2) & \widetilde{H}(3)
\end{array} \cdots$$

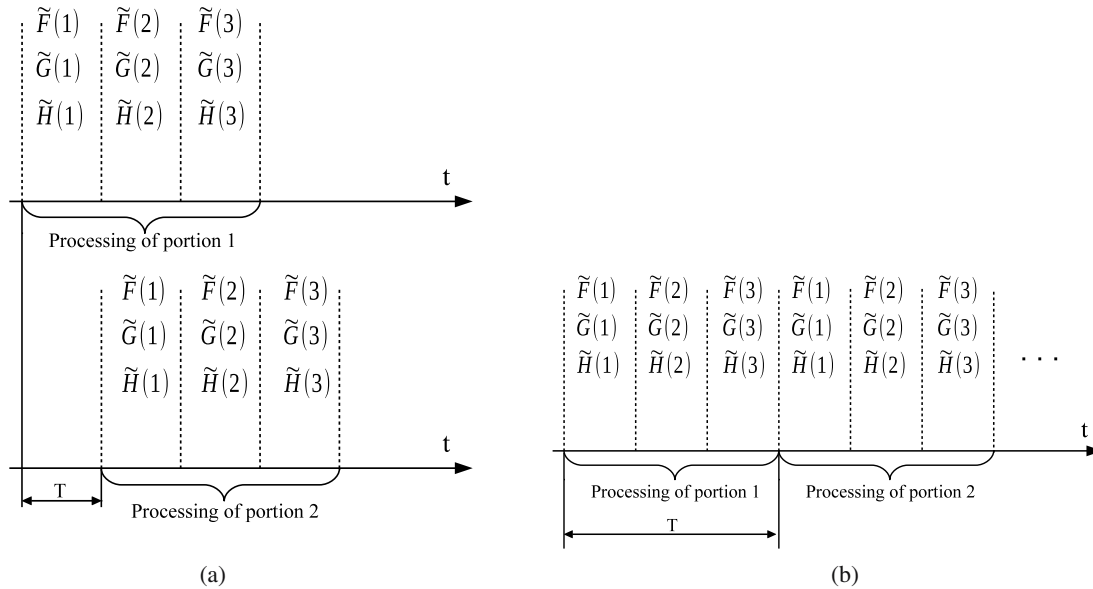Processing of portion 1 Processing of portion 2
T

(b)

Fig. 4. Transformation of a model for parallel computation (a) into a sequential periodic time-varying model (b).

whereas for a model the time position depends on the time needed to calculate it, which is usually small.

This fact is no obstacle for correct comparison of these sequences when they are represented as queues compared symbol-by-symbol without reference to times of their generation. The equality of the corresponding output vectors of sequences is explained by the identity of the algorithms for the processing of the input test vectors in a chain and in a model. The coincidence is due to the pipeline implementation of the chain, where each exchange changes only the state vectors of the interacting elements of the chain. The change only affects the vectors of the subsequent elements, but has no effect on the vectors of preceding elements. This is how each portion of input data is processed. This makes it possible to calculate the output sequence under the assumption that portions of information are processed strictly sequentially, i.e., processing the $(i + 1)$-th portion starts only after the i-th portion has been processed (Fig. 4(b)). From this it follows that, if an input sequence is a test for the chain model, it will be a test for the chain itself. ♦

Let us discuss the requirements for the models of chains and elements. From diagnosis theory (Patton *et al.*, 2000; Zhirabok, 2009) it is known that observability and controllability of a diagnosed system are the keys to effective diagnosis (monitoring). It is also true for the case under consideration, namely, when synthesizing a system model, we should ensure that this model should possess these two features. However, in our case, for a system model to be observable and controllable, it is necessary and sufficient that each of the model chains should be observable and controllable.

It has been shown above that a model of a periodically time-varying dynamic system can be applied to describe a chain in monitoring. Since the rules of synthesis for observable and controllable chains and the rules for designing tests periodically time-varying dynamic systems are known (Kolesov, 1988; 1990), they are briefly outlined in the next section.

In the meanwhile, we should call an additional requirement a chain model should meet. It is clear that any rearrangement of PMs leads to rearrangement of elements, at least, in one chain. Therefore, it will suffice to detect any rearrangement of the chain elements. Clearly, rearrangements of identical element models do not change the chain model and, consequently, in principle they cannot be detected. On the contrary, if the models of all chain elements are different, detection of rearrangements in the test is guaranteed because this leads to changes in model matrices. Therefore, it can be inferred that it is necessary to fulfill the following condition: element models corresponding to different PMs should be different to allow detection of PM rearrangements, whereas the elements forming the model of one PM can be identical.

4. Observability and controllability of a chain

First of all, it makes sense to mention a complicating circumstance. Usually, when discussing observation and control problems, we assume these processes to be continuous in time (for a discrete system, observation and control are realized in all cycles). However, in many practical situations, this condition does not hold

for the problem considered here because observation is performed only during data transmission from the output element to the MS, whereas control takes place during data transmission from the MS to input elements. Often there are only two such data exchanges on the processing interval: one for observation and the other for control. Therefore, further on we will focus on observability and controllability with respect to a limited number of observation and control points.

It is clear that such properties of any time-varying systems as observability and controllability depend on the time interval on which they are estimated. Refer to Fig. 4(b) showing a diagram of matrices for a periodically time-varying system. Generally, different periodic matrix sequences, and therefore different observability and controllability characteristics, correspond to start times of observation or control. Each sequence of matrices has its own index sequence $\Gamma = \{\gamma_r | r = \overline{1, L}\}$. Their initial segment over a period-long interval is determined by the cyclic shift of the index sequence 1, 2, …, L. When $L = 3$, we have three index sequences: $\Gamma = \{1, 2, 3; 2, 3, 1; 3, 1, 2\}$.

The system is called γ_r-controllable (observable) if it is controllable (observable) for matrix sequence γ_r. The most favourable conditions for monitoring (diagnosis) is the case when the system is controllable (observable) for any matrix sequence, i.e., completely controllable and observable. The criterion for γ_r-controllability (Kolesov, 1988) is $\operatorname{rank} P_{\gamma_r} = n$, where P_{γ_r} is γ_r-controllability matrix, with

$$
\begin{aligned}
P_{\gamma_r} = \Big[&\tilde{G}(\gamma_r(N)); \dots \tilde{F}_{n\gamma_r}^{n-1} \tilde{F}(\gamma_r(N)) \tilde{F}(\gamma_r(N-1)) \\
&\dots \tilde{F}(\gamma_r(3)) \tilde{G}(\gamma_r(2)); \\
&\tilde{F}_{p\gamma_r}^{n-1} \tilde{F}(\gamma_r(N)) \tilde{F}(\gamma_r(N-1)) \\
&\dots \tilde{F}(\gamma_r(2)) \tilde{G}(\gamma_r(1)) \Big].
\end{aligned}
$$

Here, $\tilde{F}_{p\gamma_r} = \tilde{F}(\gamma_r(N)) \tilde{F}(\gamma_r(N-1)) \dots \tilde{F}(\gamma_r(1))$ is the product of all matrices corresponding to the sequence γ_r.

The criterion for γ_r-observability (Kolesov, 1988) is $\operatorname{rank} Q_{\gamma_r} = n$, where Q_{γ_r} is the γ_r-observability matrix, with

$$
\begin{aligned}
Q_{\gamma_r} = \Big[&(\tilde{H})^T(\gamma_r(1)); (\tilde{H}(\gamma_r(2)) \tilde{F}(\gamma_r(1)))^T; \\
&(\tilde{H}(\gamma_r(3)) \tilde{F}(\gamma_r(2)) \tilde{F}(\gamma_r(1)))^T; \\
&\cdots ; \\
&(\tilde{H}(\gamma_r(N)) \tilde{F}(\gamma_r(N-1)) \tilde{F}(\gamma_r(N-2)) \\
&\dots \tilde{F}(\gamma_r(1)) \tilde{F}_{p\gamma_r}^{n-1})^T \Big].
\end{aligned}
$$

Complete observability and controllability can be provided much easier if the matrices $\tilde{F}_k, k = \overline{1, N}$ of the system model are nonsingular. Let the model be γ_i-observable and γ_j-controllable. The state x_i of

the system (at the beginning of the sequence γ_i) can be determined from the model output if the model is γ_i-observable. Each period includes only one such moment. But since the matrices $\tilde{F}_k, k = \overline{1, N}$ are nonsingular, x_i can be used to determine not only any subsequent but also any previous state. This fact is indicative of the complete observability of the system. Obviously, the same conclusion is true for controllability. Thus, we can state that a periodically time-varying system is completely observable and controllable if it is γ_i-observable, γ_j-controllable, and its matrices $\tilde{F}_k, k = \overline{1, N}$ are nonsingular.

Assume that each element of the r-th chain outputs and receives data only once within a chain schedule period. The chains can be homogenous (containing only identical elements) and heterogeneous. Two options, extreme, in a sense, are possible. First, the chain consists of elements (vector elements) with input and output vectors of maximum dimensions equal to the dimension m of the element; second, the chain consists of scalar elements. In the first, the information for monitoring transmitted within a period is maximal; in the second case, the information is minimal. In the first instance, however, observability and controllability indices (and therefore the test length) are minimum, and in the second one, these parameters are maximum.

Suppose that all elements are observable and controllable in both cases. Then the following is true.

Proposition 1. (Kolesov, 1990) *A chain composed of q vector elements of the same dimensions (h_j, f_j, g_j), $j = \overline{1, q}$ is completely observable and controllable with any order of exchanges if matrices $f_j, j = \overline{1, q}$ are nonsingular.*

The same statement as applied to a chain of scalar elements requires more stringent conditions.

Proposition 2. (Kolesov, 1990) *A chain composed of q scalar elements of the same dimensions $(h_j, f_j, g_j), j = \overline{1, q}$ is completely observable and controllable with any order of exchanges if*

1. *the characteristic polynomials $\varphi_{f_j^2}$ of matrices $f_j^2, j = \overline{1, q}$ are irreducible;*

2. *the scalar elements $(h_{1,s}, f_1^2, g_1^1), s = 1, 2;$ $(h_q', f_q^2, g_{q,t}), t = 1, 2; \{(h_{j,s}, f_j^2, g_{j,t}) | j = \overline{2, q-1}, s = 1, 2, t = 1, 2\}$, where $h_q' = [0 \dots 0 \alpha_{1q}]; h_{j,1} = [0 \dots 01]; h_{j,2} = [0 \dots 10], g_1^1 = [010 \dots 0]^T, g_{j,1} = [10 \dots 0]^T, g_{j,2} = [\alpha_{m_j-1,j} 10 \dots 0]^T, j = \overline{1, q}$ are observable and controllable.*

In the scalar case, the element requirements are stricter due to worse observability and controllability of

elements. Squared matrices $f_j \quad j = \overline{1,q}$ appear in the conditions 1 and 2 because the elements are assumed to trigger twice within a chain period. In both cases, observability and controllability are independent of the order of exchanges in the chain. It is this detail that makes the condition 2 cumbersome. This independence is critical for designing monitoring tools, since in this case inevitable transformations of the exchange order in the course of the design do not affect observability and controllability of the model.

Example 2. Consider a simple example. Let us solve the monitoring problem for the first chain of the model shown in Fig. 3. Assume that the chain contains vector binary elements.

1. Select the following element structure:

$$
f_{1,1} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad f_{1,2} = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix},
$$

$$
h_{1,1} = h_{1,2} = g_{1,1} = g_{1,2} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.
$$

2. Calculate the following matrices for reception, transmission and output of the equation for the chain:

$$
\tilde{F}_1(1) = \begin{bmatrix} f_{1,1} & 0 \\ 0 & E \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
\tilde{F}_1(2) = \begin{bmatrix} f_{1,1} & 0 \\ g_{1,2} h_{1,1} & f_{1,2} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix},
$$

$$
\tilde{F}_1(3) = \begin{bmatrix} E & 0 \\ 0 & f_{1,2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix},
$$

$$
\tilde{G}_1(1) = \begin{bmatrix} g_{1,1} \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \end{bmatrix},
$$

$$
\tilde{H}_1(3) = \begin{bmatrix} 0 & h_{1,2} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},
$$

$$
\tilde{G}_1(j) = 0, \quad j = \overline{2,3}, \quad \tilde{H}_1(j) = 0, \quad j = \overline{1,2}.
$$

It is obvious that all the dynamic matrices are nonsingular.

3. Calculate the products of all dynamic matrices within a period for all initial moments:

$$
\tilde{F}_{p,\gamma_1} = \tilde{F}_{p,\gamma_2} = \tilde{F}_{p,\gamma_3}
$$

$$
= \tilde{F}(1)\tilde{F}(2)\tilde{F}(3) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}.
$$

4. Calculate the observability matrix generally given by

$$
\begin{aligned}
Q_{\gamma_r} = [&(\tilde{H}(\gamma_r(1)))^T, \\
&(\tilde{H}(\gamma_r(2))\tilde{F}(\gamma_r(1)))^T, \\
&(\tilde{H}(\gamma_r(3))\tilde{F}(\gamma_r(2))\tilde{F}(\gamma_r(1)))^T, \\
&\cdots, \\
&(\tilde{H}(\gamma_r(N))\tilde{F}(\gamma_r(N-1))\tilde{F}(\gamma_r(N-2))) \\
&\cdots \tilde{F}(\gamma_r(1))\tilde{F}_{p\gamma_r}^{n-1})^T].
\end{aligned}
$$

However, for the chain, this expression is substantially simplified because there remain only elements containing $\tilde{H}(\gamma(N)) \; N = L + 1$. Therefore,

$$
\begin{aligned}
&Q_{\gamma_r} \\
&= [(\tilde{H}(\gamma_r(N))\tilde{F}(\gamma_r(N-1))\dots\tilde{F}(\gamma_r(2))\tilde{F}(\gamma_r(1)))^T, \\
&\quad \cdots, \\
&\quad (\tilde{H}(\gamma_r(N))\tilde{F}(\gamma_r(N-1))\tilde{F}(\gamma_r(N-2)) \\
&\quad \cdots \tilde{F}(\gamma_r(1))\tilde{F}_{p\gamma_r}^{n-1})^T].
\end{aligned}
$$

In this case, $N = 3$ (the number of exchanges), $n = 4$ (the dimension of the chain model state vector). Then, for $\gamma_1 = 1, 2, 3$,

$$
\begin{aligned}
Q_{\gamma_1} = [&(\tilde{H}(\gamma_1(3))\tilde{F}(\gamma_1(2))\tilde{F}(\gamma_1(1)))^T; \\
&(\tilde{H}(\gamma_1(3))\tilde{F}(\gamma_1(2))\tilde{F}(\gamma_1(1))\tilde{F}_{p\gamma_1})^T; \\
&(\tilde{H}(\gamma_1(3))\tilde{F}(\gamma_1(2))\tilde{F}(\gamma_1(1))\tilde{F}_{p\gamma_1}^2)^T] \\
= [&(\tilde{H}(3)\tilde{F}(2)\tilde{F}(1))^T; \\
&(\tilde{H}(3)\tilde{F}(2)\tilde{F}(1)\tilde{F}_{p\gamma_1})^T; \\
&(\tilde{H}(3)\tilde{F}(2)\tilde{F}(1)\tilde{F}_{p\gamma_1}^2)^T] \\
= &\begin{bmatrix} 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix}.
\end{aligned}
$$

Here, the first two elements were sufficient to obtain a complete set of linearly independent columns of the observability matrix, i.e., the system observability index is 2. Thus, the chain is γ_1-observable.

Similar operations show that the chain is completely observable.

5. Calculate the controllability matrix generally given by

$$P_{\gamma_r} = [\tilde{G}(\gamma_r(N));$$
$$\dots \tilde{F}_{p\gamma_r}^{n-1}\tilde{F}(\gamma_r(N))\tilde{F}(\gamma_r(N-1))$$
$$\tilde{F}(\gamma_r(3))\tilde{G}(\gamma_r(2));$$
$$\tilde{F}_{p\gamma_r}^{n-1}\tilde{F}(\gamma_r(N))\tilde{F}(\gamma_r(N-1))$$
$$\dots \tilde{F}(\gamma_r(2))\tilde{G}(\gamma_r(1))].$$

However, for the chain considered, this expression is substantially simplified because there remain only elements containing $\tilde{G}(\gamma(1))$. Therefore, for $\gamma_1 = 1, 2, 3$,

$$P_{\gamma_1} = [\tilde{F}(\gamma_1(3))\tilde{F}(\gamma_1(2))\tilde{G}(\gamma_1(1));$$
$$\tilde{F}_{p\gamma_1}\tilde{F}(\gamma_1(3))\tilde{F}(\gamma_1(2))\tilde{G}(\gamma_1(1));$$
$$\tilde{F}_{p\gamma_1}^2\tilde{F}(\gamma_1(3))\tilde{F}(\gamma_1(2))\tilde{G}(\gamma_1(1))]$$
$$= [\tilde{F}(3)\tilde{F}(2)\tilde{G}(1);$$
$$\tilde{F}_{p\gamma_1}\tilde{F}(3)\tilde{F}(2)\tilde{G}(1);$$
$$\tilde{F}_{p\gamma_1}^2\tilde{F}(3)\tilde{F}(2)\tilde{G}(1)]$$
$$= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}.$$

Here, the first two elements were sufficient to obtain a complete set of linearly independent columns of the controllability matrix, i.e., the system controllability index is 2. Thus, the chain is γ_1-controllable. Similar operations show that the chain is completely controllable. $\blacklozenge$

5. Design of a test for a periodically time-varying model of a chain

Let us briefly discuss the design of a fault detection test for a periodically time-varying system. The test U_T consists of $L + 1$ pairs $U_{\gamma_r} = (U_{1,\gamma_r}, U_{2,\gamma_r})$ of fragments $U_T = U_{\gamma_1}U_{\gamma_2}\dots U_{\gamma_{L+1}}$ (Kolesov, 1988). The nominal reaction of the system at an r-th pair indicates the absence of distortions in the matrices $\tilde{F}(\gamma_r(1)), \tilde{G}(\gamma_r(1)), \tilde{H}(\gamma_r(1)), r = \overline{1, L+1}$.

In the first fragment U_{1,γ_r} of the pair, which confirms or refutes the absence of distortion in matrix $\tilde{F}(\gamma_r(1))$, the system model in state space passes the states of some arbitrary selected basis $\{x_i | i = \overline{1, n}\}$. For each state x_i, the fragment includes homing sequence $u_{\gamma_r}^{i*}$ and free motion interval (at the input of system of $n(L+1)$ zeroes denoted as $0^{n(L+1)}$).

Thus, the first fragment of the pair has the following structure:

$$U_{1\gamma_r} = u_{1\gamma_r}^* 0^{n(L+1)} u_{2\gamma_r}^* 0^{n(L+1)} \dots u_{n\gamma_r}^* 0^{n(L+1)}. \quad (5)$$

In the second fragment U_{2,γ_r} of the pair, which confirms or refutes the absence of of distortion in

matrices $\tilde{G}(\gamma_r(1))$ and $\tilde{H}(\gamma_r(1))$, vectors $u_{i,\gamma_r}, i = \overline{1, m}$ consisting of zero components, except for one component equal to 1 in the i-th bit, are fed to the system input sequentially. After each vector, the system is in free motion on $n(L + 1)$ cycles. Therefore, the second fragment of the pair U_{γ_r} has the following structure:

$$U_{2\gamma_r} = u_{1\gamma_r}^* 0^{n(L+1)} u_{2\gamma_r}^* 0^{n(L+1)} \dots u_{m\gamma_r}^* 0^{n(L+1)}. \quad (6)$$

Example 3. Design the first pair $U_{\gamma_r} = (U_{1,\gamma_1}, U_{2,\gamma_1})$ of test fragment for the first chain from Example 2. Design the first fragment U_{1,γ_1} of the pair.

1. Generate a basis in the state space:

$$x^1 = \begin{bmatrix} x_1^1 \\ x_2^1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix},$$

$$x^2 = \begin{bmatrix} x_1^2 \\ x_2^2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix},$$

$$x^3 = \begin{bmatrix} x_1^3 \\ x_2^3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix},$$

$$x^4 = \begin{bmatrix} x_1^4 \\ x_2^4 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},$$

where x_i^j is the state vector of an i-th element in a j-th vector of the basis.

2. Generate the homing sequences for these states on the sequence of matrices γ_1, that is to say, on the sequence of matrices with initial segment $(\tilde{F}_1(1), \tilde{G}_1(1), \tilde{H}_1(1)), (\tilde{F}_1(2), \tilde{G}_1(2), \tilde{H}_1(2)), (\tilde{F}_1(3), \tilde{G}_1(3), \tilde{H}_1(3)),$

$$u^{1*} = P_{\gamma_r}^{-1}[x(nN) \oplus \tilde{F}_{p\gamma_r}^n x(0)]$$
$$= P_{\gamma_1}^{-1}[x(12) \oplus \tilde{F}_{p\gamma_1}^2 x(0)]$$
$$= \begin{bmatrix} u_{1,1}^1 \\ u_{1,2}^1 \\ u_{2,1}^1 \\ u_{2,2}^1 \end{bmatrix}$$
$$= \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \oplus \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\}$$
$$= \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}.$$

Here, $\oplus$ denotes the sum in the binary field (modulo 2 sum), $u_{k,l}^1$ is the value of the l-th input in the k-th element of the first homing sequence u^{1*}, and the matrix inverse to the controllability matrix is given by

$$P_{\gamma_1}^{-1} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}.$$

The test fragment following u^{1*} will be a sequence of 8 zeros (free motion), which brings the model to some final state x_k^1. Then, we form homing sequence u_2^* from this state to the second state of the basis x^2, etc.

Design the second fragment U_{2,γ_1} of the pair.

3. Form the test vectors

$$u_{1,\gamma_1} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad u_{2,\gamma_1} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

After each of them, the sequence of 12 zeros is placed in accordance with (6) (free movement). ♦

6. Conclusions

This paper discusses the monitoring of parallel computations. It is assumed that computations are implemented by a set of program modules and are based on an asynchronous organization. The monitoring procedure falls into a set of algorithms, each of them matched with a computation path in the program information graph. It was shown that the model of a periodically time-varying dynamic system can be used for monitoring. A test design algorithm for this model was described.

References

Cabasino, M.P., Giua, A. and Seatzu, C. (2011). Discrete event diagnosis using labeled Petri nets: An application to manufacturing systems, *Control Engineering Practice* **19**(9): 989–1001.

Chanthery, E. and Pencole, Y. (2009). Monitoring and active diagnosis for discrete-event systems, *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Techinal Process, Barcelona, Spain*, pp. 1545–1550.

Cheng, A.M.K. (2002). *Real-Time Systems: Scheduling, Analysis and Verification*, Wiley-Interscience Publication, Inc., Hoboken, NJ.

Kolesov, N.V. (1988). Diagnosis of linear discrete-time nonstationary systems, *Automation and Remote Control* **49**(7): 943–948.

Kolesov, N.V. (1990). Nonstationary diagnostic model of communication system in a distributed control complex, *Automation and Remote Control* **51**(4): 540–549.

Lefebvre, D. (2012). Diagnosis with Petri nets according to partial events and states observation, *Proceedings of the 8th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes, Mexico City, Mexico*, pp. 1244–1249.

Patton, R.J., Frank, P.M. and Clark, R.N. (2000). *Issues in Fault Diagnosis for Dynamic Systems*, Springer-Verlag, London.

Peterson, J.L. (1981). *Petri Net*, Prentice-Hall, Englewood Cliffs, NJ.

Wu, Y. and Hadrjicostis, C. (2005). Algebraic approaches for fault identification in discrete-event systems, *IEEE Robotics and Automation* **50**(12): 2048–2053.

Zhirabok, A. N. (2009). Robust observer design: Logic-dynamic approach, *Proceedings of the 7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Process, Barcelona, Spain*, pp. 768–791.

Alexander M. Gruzlikov graduated from the National Research University of Electronic Technology (MIET) in computer science in 1999. Since 2006 he has worked at the public corporation Concern CSRI Elektropribor, as the chief of the research department. His research interest include problems of large-scale computer systems, software development and data mining applications. He has co-authored more than 10 publications.

Nikolai V. Kolesov graduated from Leningrad Electrotechnical Institute, specializing in radio engineering, in 1969. He received the Ph.D. in the diagnosis of specialized computer systems in 1977 from same institute. Since 1985 he has been working at the public corporation Concern CSRI Elektropribor as the chief researcher. He received the degree of a doctor of technical sciences in the diagnosis of specialized computer systems in 1992. His research interests include the problems of developing navigation systems and real-time computing systems, including planning, computing, diagnosis, application of artificial intelligence methods. He has co-authored more than 150 publications.

Marina V. Tolmacheva graduated from the Leningrad Institute of Exact Mechanics and Optics, specializing in electronic computers. Since 2002 she has been working at the public corporation Concern CSRI Elektropribor, first as the leading engineer and then as a senior researcher. She received the Ph.D. degree in scheduling and monitoring of computing process in sea navigating complexes in 2007. Her research interests include the problems of development of computing systems of real time, including the planning of calculations, diagnosis, application of methods of an artificial intellect. She has co-authored more than 50 publications.