Functional and test diagnosis of onboard systems using dynamic models

E V Lukoyanov¹, A M Gruzlikov¹ and N V Kolesov¹

¹Concern CSRI Elektroprivor, JSC, Saint Petersburg, Russia

E-mail: lukoyanov.egor@mail.ru

Abstract. The paper discusses the problems of troubleshooting in onboard computing and control systems. Herewith the approaches based on the methods of functional and test diagnostics are analyzed. Both approaches involve the introduction of redundancy, which in fact is a dynamic software model of the systems being diagnosed. In the case of functional diagnostics, a full-fledged model is used that forms the basis for diagnostic observers; in the case of test diagnostics, it is a simplified event model.

Introduction

The problems of diagnosis occupy a significant place in the process of designing onboard computing of integration navigation complex and control systems [1, 2]. It is known that these problems are closely related to ensuring fault tolerance, and, therefore, the reliability of the object. The paper deals with the problems of troubleshooting taking into account the features of the presented class of systems, including: high dimensionality of a system, parallel and asynchronous nature of calculations, uncertainties in object description models, the multiplicity of failure reasons (hardware, software, and errors in the organization of calculations).

Solving the problems of technical systems diagnostics is carried out using functional and test diagnostic techniques [3]. The paper analyzes the approaches that correspond to both techniques and are based on the synthesis of the system model, which is the introduced redundancy and belongs to the diagnostic tools.

Thus, the paper proposes an algorithm for synthesizing functional diagnostics tools on the basis of a bank of interval observers to solve the problem of searching for hardware faults in onboard control systems. We also propose an algorithm for synthesizing test diagnostics tools based on a parallel event model approach to solve the problem of detecting failures associated with violations in addressing intermodule exchanges in onboard distributed computing systems.

1. Functional diagnosis of systems in the presence of model parameter uncertainties

Below we examine the case of functional diagnostics, and the presentation focuses on the use of so-called state observers. The control system components are assumed to be the objects of diagnostics. In the scientific literature, much attention is paid to the problem of state observers constructing [4-7], first of all, as an integral part of any control system that solves the problem of evaluating the object's state vector. Then, based on this consideration, the necessary control is formed. The purpose of the observers discussed below is different, namely, troubleshooting in the control system hardware, but the problems facing the developer in both cases have much in common. The most acute among them is the problem of adequacy of the used model of the system under diagnosis or a control object in the case of solving
the control problem. Unfortunately, in most cases, the system model is not fully known to the developer, which significantly complicates the solution of diagnostics and evaluation tasks. One of the well-known approaches to overcoming this problem is the use of interval observers [8-10]. Further discussion is devoted to the development of diagnostic tools based on a Bank of interval observers in order to improve the efficiency of solving the problem of troubleshooting in dynamic systems.

1.1. Problem statement
Suppose that there is a continuous, linear, stationary dynamic system with one input and output, which can be described by a system of equations:

\[
\dot{x}(t) = Fx(t) + d(t) \\
y(t) = Hx(t) + v(t),
\]

(1)

where \( x(t) \) – state vector, \( F \) – dynamics matrix, \( d(t) \) – vector of disturbances, \( y(t) \) – output signal, \( H \) – output matrix, \( v(t) \) – measurement noise. A gyro stabilized platform, an autopilot, an engine, etc., can be considered as an example of such a dynamic system.

Even if the vectors \( d(t) \) and \( v(t) \) are not defined, they have guaranteed intervals of values for \( d(t) \leq \bar{d}(t) \leq \underline{d}(t) \) and \( |v(t)| \leq V \) for \( \forall t \geq 0 \), in addition, the initial conditions \( x(0) \) are unknown and belong to the guaranteed interval \( \underline{x}_0 \leq x(0) \leq \bar{x}_0 \) for \( \forall t \geq 0 \).

Basing on these assumptions, it is required to synthesize the bank of Luenberger observers for the system (1), as well as the bank of interval observers \( \{O_i\}_{i=1}^{N} \) , and compare the efficiency of solving the problem of fault search in both cases.

1.2. Synthesis of observers
When synthesizing the Luenberger observer [4] for system (1), the components \( d(t) \), \( v(t) \) are ignored. Thus, the observer is defined using the system of equations:

\[
\dot{\hat{x}}(t) = (F - LH)\hat{x}(t) + Ly(t) \\
\hat{y}(t) = H\hat{x}(t)
\]

(2)

where \( \hat{x}(t) \) is the estimate of the state vector, \( \hat{y}(t) \) – is the output signal of the observer, and \( L \) – is the feedback gain matrix for discrepancy. The matrix of the observer's dynamics \( (F - LH) \) is selected in such a way that the observer (2) is stable.

The interval observer is prescribed by a system of differential equations, taking into account the restrictions imposed on the disturbing effect and measurement noise:

\[
\dot{\underline{x}}(t) = (F - LH)\underline{x}(t) + Ly(t) - [L][E]V + \underline{d}(t) \\
\dot{\bar{x}}(t) = (F - LH)\bar{x}(t) + Ly(t) + [L][E]V + \bar{d}(t),
\]

(3)

where \( \underline{x}(t) \), \( \bar{x}(t) \) – the estimate of the lower and upper bounds of the state vector, \( E \) – the unit matrix, and \( [\underline{d}(t), \bar{d}(t)] \) and \( [-V, V] \) – the intervals for perturbation and measurement noise, respectively. The matrix of observer dynamics \( (F - LH) \) for ensuring the stability of the observer and fulfilling the condition:

\[
\underline{x}(t) \leq x(t) \leq \bar{x}(t),
\]

must satisfy the Hurwitz criterion and the Metzler criterion [8].

1.3. Decision-making function
To solve the problem of troubleshooting, we use the algorithm proposed in [13], the main feature of which is the assumption that there is no clear boundary between the functional and non-functional technical states (TS) of the system. Note that in some cases, the algorithm should be used in conjunction with the bank of interacting observers, which can increase the sensitivity to small faults.

Let us clarify the fuzzy technical state of an object by parameter $\Theta$ as a linguistic variable characterized, for example, by two terms (fuzzy sets) - a functional vehicle and a non-functional vehicle, which are described by the corresponding membership functions $\mu_i^0$ and $\mu_i^1$. The value domains of the determining parameter $\Theta$ that correspond to the object's functional and non-functional vehicle intersect and are described by the corresponding membership functions with parameters "a" and "b". As a result, for any value of the determining parameter $\Theta = \Theta_i$, the technical state of the object can be correlated with either a fuzzy set of functional states or a fuzzy set of non-functional states. Note that in the present paper, consideration is limited to the use of trapezoidal membership functions:

$$
\mu_i^0 = \begin{cases} 
1, & 0 \leq \Theta_i \leq a \\
\frac{b - \Theta_i}{b - a}, & a \leq \Theta_i \leq b \\
0, & b < \Theta_i
\end{cases},
\quad
\mu_i^1 = \begin{cases} 
0, & 0 \leq \Theta_i \leq a \\
\frac{\Theta_i - a}{b - a}, & a \leq \Theta_i \leq b \\
1, & b < \Theta_i
\end{cases}.
$$

The rule of fault decision making is based on the notion of the confidence coefficient $K_i^*$ of the $i$ th technical state introduced below. It requires that the confidence coefficient $K_i^*$ should reach a specified level $A$ for a technical state with the dominant value of this coefficient; i.e. the following condition should be satisfied:

$$
K_i^* = \max_i \{ K_i \} \geq A.
$$

Let us clarify the procedure of calculation of confidence coefficients. It is based on residuals $\{ \nu_i \}_{i=0}^{N}$ that determine the technical state of the dynamic system. Residuals are formed when comparing the outputs of the system with the outputs of observers. It is assumed that the residual $\nu_i$ formed in a result of comparison of the system outputs with the $i$ th observer, can be represented by a linguistic variable, for example, with two terms - "small" and "large", for which the membership functions $\mu_{i,0}$ and $\mu_{i,1}$, $i = 0, N$, are defined.

In order to obtain confidence coefficients $\{ K_i \}_{i=0}^{N}$, first, we determine the characteristics called the generalized membership degrees $\{ \mu_i \}_{i=0}^{N}$ of the technical state of the diagnosed system to each of the possible fuzzy technical states. These characteristics generalize the information about the technical state of the system with respect to all observers and are formed based on sets of values $\{ \nu_i \}_{i=0}^{N}$. The value of the generalized membership degree $\bar{\mu}_i$ is formed in accordance with the following expression:

$$
\bar{\mu}_i = \mu_{i,0} \prod_{j=0}^{N} \mu_{i,1}.
$$
Indeed, the observer adequate to the technical state of the system forms a small residual, whereas the others form large residuals. For now, we do not consider situations including equivalent or poorly distinguishable faults.

Then the confidence coefficient $K_j$ for each technical state is calculated in accordance with the rule of weighting coefficients by determining the contribution of the generalized membership degree $\tilde{\mu}_i$ to the sum of these degrees for all states:

$$K_j = \frac{\tilde{\mu}_i}{\sum_{j=0}^{N} \tilde{\mu}_j}.$$  \hspace{1cm} (4)

1.4. Modelling

As an example, consider the dynamic model of a DC motor, which is described by a system of second-order differential equations:

$$J \ddot{\theta} + b \dot{\theta} = K_e i + d_1(t)$$
$$L \frac{di}{dt} + Ri = V - K_t \dot{\theta} + d_2(t),$$

where $\theta$ – angle of the rotor rotation, $i$ – current strength, $J$ – generalized moment of inertia of the rotor, $b$ – viscous friction coefficient, $K_e$ – coefficient of back EMF, $L$ – inductance, $R$ – electrical resistance, $V$ – input voltage, $K_t$ – motor torque coefficient. In our case, $K = K_e = K_t$, since they are equivalent in SI units. To demonstrate the effectiveness of the proposed solutions based on the bank of interval observers (3), we will compare their performance with solutions based on the use of the bank of classical Luenberger observers (2). Periodic jump-like increase of $R$ parameter was modeled as a fault. Fig. 1 shows the graphs of confidence coefficients (4) that determine the technical state of the system using the bank and Luenberger observers (figure 1a) and interval observers (figure 1b) at the maximum level of external disturbance.

As a result, after a series of experiments with different values of the fault level and the disturbance effect level $d_1(t)$ and $d_2(t)$, the probability estimate was obtained for misidentification of the
technical state of the system, which showed that the use of the bank of interval observers, especially in
the case of “great” faults, correctly determines the technical state with an error less than 5%, whereas in
the case of using the bank of Luenberger observers the diagnostics tools work with an error exceeding
30%.

2. Test diagnostics of systems using a parallel model
The object of diagnostics for further consideration is an arbitrary onboard distributed real-time
computing system (RTC), represented by a set of functionally related software modules. For RVS, the
development of diagnostic tools (DT) is a rather complex problem due to high dimensionality of the
system and multiplicity of causes of violations. Not only hardware faults can be the source of violations,
but also errors in organization of calculations and in the programs used, made by developers. Within the
framework of test diagnostics, one of the possible approaches to mathematical description of the
considered class of systems is the use of a so-called parallel model [14, 15]. The model is embedded in
the system software, executed in parallel with the main functional algorithms, and is intended for test
diagnostics of violations in addressing exchanges between the system's program modules (PM). An
approach based on finite-state-machine models is also known [16, 17]. However, in this case, the
asymptotic complexity of the system test construction algorithm, which determines the relation between
its execution time increase and the model dimension, is characterized by the exponential dependence on
the model dimension. In the case of using a parallel model, the developer chooses the algorithms for
processing test sequences in each PM, and does so in such a way as to simplify both the algorithm for
constructing the test and the test itself, namely, he selects a linear model that allows using algorithms
for constructing the tests with asymptotic complexity characterized by the polynomial dependence on
the model dimension. Reducing the complexity of the used algorithms for constructing the tests becomes
especially important in the case of RTC.

Let’s illustrate the procedure for constructing the structure of a parallel model on a simple example
(figure 2a). Here we can see the system defined by the graph of intermodule links. The system
implements three functionally related PM: PM₁, PM₂ and PM₃ which can be located either on different
processors of the system or on the same one. Each PM generates the output data ( y₁, y₂ and y₃,
respectively) based on the input data ( μ₁, μ₂ and y₃ – for PM₂, y₁, and y₂ – for PM₃). The input data are received
and processed in real time with a certain specified period. In the system, all input streams of a particular PM
are arguments of the function it implements, which are necessary for its calculation.

Figure 2b shows the system together with the parallel model and the DT. It is worth noting that the
model is the introduced redundancy for the system and is represented by diagnostic algorithms: π₁, π₂,
and π₃. Diagnostic tools are implemented as a separate program module, highlighted with a dashed
line, and consist of a test generator (GT), a reference response generator (GRR), and a comparator (K).
The test data μ₁' and μ₂' generated by the DT complement the input data for the new PM₁', and PM₂',
also highlighted by the dashed line and resulting from combining the original PM with diagnostic
algorithms. Thus, each PMₐ is via exchange channels receives information that is processed by
standard algorithms. As an example of a standard algorithm we can use processing of a stream of digital
data from an array of sensors or solving of the orientation and navigation problems by integrated
complexes [16–18], which are generally nonlinear. In parallel with this (hence the name of the model),
test data are processed by diagnostic algorithms reacting to the information reception/output events.
Since the mechanism of real yₐ and test yₐ data exchange in the system is common, the analysis
of test results serves as a basis for the conclusion about whether there are any faults in the addressing of
exchanges between the PMs or not. Note that distortions of real data in the exchange process while
saving the graph of information links are not included in this class of violations being considered.
Thus, the process of parallel model synthesis consists in constructing diagnostic algorithms for processing the test sequences. Further discussion is devoted to the development of diagnostic tools, in particular, algorithms for constructing tests for a parallel model in order to detect a class of faults caused by violations in the addressing the intermodule exchanges.

2.1. Statement of the problem

One of the possible variants of the parallel model is a model with independent chains [19]. In this case, at the first stage, the model structure is formed, which is a set of computational paths (routes) that cover all edges of the graph of intermodule links of the initial system. At the same time, we understand the computational path as a sequence of triggered PM connecting a certain input to the output. Then each of the obtained paths is compared to the circuit \( l_j = \{v_i\}_{i=1}^{n_j} \) of such number of dynamic links \( v_i \), through how many PM this path passes; \( n_j \) – the total number of links in the \( j \)th chain. Thus, the model structure is a set of independent chains \( \{l_j\}_{j=1}^{m} \), where \( m \) is the total number of independent chains in the model. Figure 3a shows the dynamic links \( v_{i,j} \), where \( i \) is the link number in the chain, \( j \) is the chain number. At the second stage of model construction, the type of dynamic links is determined. Here we take into account that the desired dynamic model of the system is used further for designing the tests and that the procedure for designing tests is simplified if the system model is, first, linear, and second, controllable and observable [5]. From here, we can formulate a requirement for dynamic links – they must be linear, i.e.

\[
x'_{i,j}(k+1) = f_{i,j}x'_{i,j}(k) + g_{i,j}u'_{i,j}(k), \quad y'_{i,j}(k) = h_{i,j}x'_{i,j}(k), \quad i = \overline{1,n_j}, \quad j = \overline{1,m}, \quad (5)
\]
where \( \mathbf{x}'_{i,j}(k) \in \mathbb{F}^n \), \( \mathbf{u}'_{i,j}(k) \in \mathbb{F}^q \), \( \mathbf{y}'_{i,j}(k) \in \mathbb{F}^p \) – vectors of state, input and output, respectively, for the \( i \)-th level model of the \( j \)-th chain in the Galois field \( \mathbb{F} \); \( n \) – state vector dimension; \( q \) – input vector dimension; \( P \) – output vector dimension, \( f_{i,j} \in \mathbb{F}^{n \times q} \), \( g_{i,j} \in \mathbb{F}^{p \times q} \), \( h_{i,j} \in \mathbb{F}^{p \times p} \) matrixes of dynamics, input and output, respectively. In addition, the links must provide observability and controllability of the system model.

The dynamic description of a chain consisting of links is obtained by the following rules. The state vector \( \mathbf{x}'_i(k) \) of the chain is formed from the state vectors of the links \( \mathbf{x}'_{i,j}(k) \) included in it. Let us assume that only one exchange takes place in the system at any time, then the information transfer between PM and DT can be described using block matrices \( \mathbf{F}_j(k) \), \( \mathbf{G}_j(k) \), \( \mathbf{H}_j(k) \), made up of the dynamics matrices, input and output of the links described by the model (5). In practice, this assumption is not fulfilled, however, as shown in [3, 20, 21], this is not an obstacle to using such models in the tests designing. For ease of description, let us relate each sequence of matrices at an interval, equal to the period, with its own sequence of indices, obtained from the set \{1, 2, ..., \} \( \mathbb{F} \) as a result of cyclic shift. For example, for \( N=3 \) we get a set consisting of three sequences of indices \( \Gamma = \{1, 2, 3; 2, 3, 1; 3, 1, 2\} \).

Let us denote an element of the set \( \Gamma \) as \( \gamma_s \). Then

\[
\mathbf{x}'_i(k+1) = \mathbf{F}_j(\gamma_s(k))\mathbf{x}'_i(k) + \mathbf{G}_j(\gamma_s(k))\mathbf{u}'_i(k), \quad \mathbf{y}'_i(k) = \mathbf{H}_j(\gamma_s(k))\mathbf{x}'_i(k), \quad k = 1, N, \quad (6)
\]

where \( \mathbf{u}'_i(k) \) is the vector of input actions of the chain, \( \mathbf{y}'_i(k) \) is the vector of the output response of the chain, \( N \) – the number of exchanges during the system operation period. Thus, the chain model is described using a periodically non-stationary linear dynamic system.

In terms of model (6), the class of violations under consideration is defined as all possible distortions of the matrices of this model. Basing on these assumptions, we need to design a test that is an input action for the model that detects the specified class of faults.

2.2. Test designing algorithm for a parallel model with independent chains

The verification test for the chain model (6) consists of \( N \) fragments:

\[
\mathbf{U}^{\text{frg}} = \mathbf{U}_{\gamma_1} \mathbf{U}_{\gamma_2} \ldots \mathbf{U}_{\gamma_N},
\]

each of them includes two inherent parts:

\[
\mathbf{U}_{1\gamma_s} = \mathbf{u}_{1,\gamma_s}^* \mathbf{0}^N \mathbf{u}_{2,\gamma_s}^* \mathbf{0}^N \ldots \mathbf{u}_{n,\gamma_s}^* \mathbf{0}^N, \quad \mathbf{U}_{2\gamma_s} = \mathbf{u}_{1,\gamma_s}^* \mathbf{0}^N \mathbf{u}_{2,\gamma_s}^* \mathbf{0}^N \ldots \mathbf{u}_{m,\gamma_s}^* \mathbf{0}^N.
\]

In all the fragments of \( \mathbf{U}_{1\gamma_s} \), the system goes through the states of a certain selected basis \( \mathbf{X} = \{\mathbf{x}_r\}_{r=1}^n \) in the state space for a sequence of matrix indices \( \gamma_s \). For each state \( \mathbf{x}_r \), the fragment contains a homing sequence \( \{\mathbf{u}_{r,\gamma_s}^*\}_{r=1}^m \) of a length that is a multiple of \( N \) and a free motion interval for the sequence of matrix indices \( \gamma_s \), when a sequence of \( nN \) zero vectors is fed to the model input, denoted as \( \mathbf{0}^{nN} \). Due to the fact that the lengths of the homing sequences are multiples of \( N \), the system passes through all the intervals of free motion with the sequence of matrices \( \gamma_s \). In all the fragments of \( \mathbf{U}_{2\gamma_s} \), vectors \( \{\mathbf{u}_{r,\gamma_s}\}_{r=1}^m \) belonging to a certain basis of the space of input vectors are sequentially fed to the model.
input. After each vector, the model is in free motion on \( nN \) cycles with a sequence of matrix indexes \( \gamma_s \).

Further, when designing the test the following well-known fact is used: for any chain described by model (6) and having during the operation period only one session of information reception from the diagnostic tool, described for a sequence \( \gamma_s \) of indices by matrices \( F(\gamma_s(N)), G(\gamma_s(N)), H(\gamma_s(N)) \), and one session of information delivery to diagnostic tools described by the matrices \( F(\gamma_s(N-1)), G(\gamma_s(N-1)), H(\gamma_s(N-1)) \), there exists a stationary system:

\[
\begin{align*}
    x(k+1) &= Ax(k) + Bu(k), \\
    y(k) &= Cx(k),
\end{align*}
\]

where

\[
A = F_p(\gamma_s) = \prod_{i=1}^{N} F(\gamma_s(N-i+1)), \quad B = G(\gamma_s(N)), \quad C = H(\gamma_s(N-1))F^{-1}(\gamma_s(N-1))F^{-1}(\gamma_s(N))A = H(\gamma_s(N-1))A,
\]

which, for any input sequence, generates output sequences that coincide with the output sequences of the chain on \( \gamma_s \), here \( N \) – periods of the chain model (6).

It is known [14] that the test \( U' \) for the stationary model (7) consists of two fragments \( U'_1 \) and \( U'_2 \). At the same time it is also a test for a periodically non-stationary model of the chain (6), but it is characterized by a significantly shorter length than \( U^{ns} \), and a guaranteed list of detectable faults that matches the corresponding list for \( U' \).

2.3. Example of test designing

Let's design a test for a model of a chain consisting of two links (figure 3a). Let us denote a sequence of matrix indices, at which the transition from a non-stationary representation of the model (6) to a stationary one (7) is possible, as \( \gamma_i \in \Gamma \). A special feature of this sequence of sessions is location of the output and input sessions at the end of the period.

![Figure 3](image-url)

**Figure 3.** Chain consisting of two links (a), distortion in the session of data reception from the DT (b).

Let us calculate the matrices \( A, B, C \), for the model (7). In this case, the matrix \( A \) is defined as a product of dynamics matrices of all sessions on a period with the sequence of matrices \( \gamma_1 \):
\[ A = F(\gamma(3))F(\gamma(2))F(\gamma(1)) = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix}. \]

For the matrix we have \( B = G(\gamma(3)) \), and for the matrix \( C \)

\[ C = H(\gamma(2))A = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{bmatrix}. \]

The controllability matrix is calculated as:

\[ P = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \]

Let's design a test for the generated model \( U^s = U^s_1U^s_2 \). To start, let's form a basis in the state space:

\[ x_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad x_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad x_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad x_4 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \]

Let's construct the first part \( U^s_1 \), of the test, which contains the homing sequences for these states, using the expression from [3]. In this case, for the first homing sequence, we have:

\[ u^*_1 = P^*[x_T(n) \oplus A^n x(0)] = \begin{bmatrix} u^*_{1,2} \\ u^*_{1,1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 & 0 & 0 & 0 \\ 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \]

Here, the symbol \( \oplus \) denotes the sum in the binary field (modulo sum of 2), \( u^*_{1,k} \) – homing sequence at the \( k \)-th step of the first homing sequence \( u^*_1 \). The test fragment following \( u^*_1 \) is a sequence of vectors containing zeros (free motion), which brings the model to a certain final state \( x_1 \). Next we design homing sequence \( u^*_2 \) to bring the model from this state to the second state of basis \( x_2 \), and so on.

Let's design the second fragment \( U^s_2 \) of the test. Form test vectors

\[ u_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad u_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \]

Each of them is followed by a sequence of zero vectors (free movement).
Let's illustrate the detecting ability of the designed test. Let the distortion be represented as the case when in the session of data reception from DT the test information comes not only to the first \( v_{1,1} \), but also to the second \( v_{1,2} \) link (Fig. 3b), which corresponds to the following distortion (highlighted in underline) in the matrices of the sharing session \( F(\gamma (3)) \) and \( G(\gamma (3)) \):

\[
\tilde{F}(\gamma (3)) = \begin{bmatrix} f_{1,1} & 0 \\ 0 & f_{2,1} \end{bmatrix}, \quad \tilde{G}(\gamma (3)) = \begin{bmatrix} g_{1,1} \\ g_{2,1} \end{bmatrix}.
\]

This distortion will be detected when the first part of the test is applied to the circuit input, transferring the model from the initial state to the first state from the basis \( x_1 \). This fact follows from Table 1, which shows the initial fragment of the input test sequence, the corresponding output reference sequence, and the actual response of the faulty system to this fragment.

### Table 1. Fragment of test, reference sequence and system response.

| Test sequence | Reference sequence | Response in the presence of a fault in the circuit |
|---------------|--------------------|--------------------------------------------------|
| 0 1 0 0       | 0 0 1 1            | 0 1 1 1                                          |
| 0 0 0 0       | 0 0 0 0            | 0 0 0 0                                          |

**Conclusion**

The paper discusses possible approaches to troubleshooting in onboard computing and control systems. Both functional and test diagnostics techniques are considered.

As a part of the first technique, it is proposed to use banks of interval observers. Using a specific example, we compare the effectiveness of solving the fault search problem based on the bank of Luenberger observers, as well as the bank of interval observers. The simulation results showed that the use of the interval observer bank, especially in the case of "great" faults, correctly determines the technical state with an error of less than 5%, whereas in the case of using the Luenberger observer bank, the diagnostic tools work with an error of more than 30%.

The second approach considers the process of synthesizing a parallel diagnostic model of an onboard computer system. The procedure for synthesizing the test for a model has been developed, which detects all sorts of violations in addressing intermodule exchanges, leading to non-equivalent distortions of the matrices of this model. The example demonstrates detectability of the test.

**Acknowledgments**

The work was supported by the RFBR grant 19-08-00052.

**References**

[1] Paramonov P P and Zharinov I O 2013 Integrated On-board Computing Systems: Present Situation Review and Development Prospects Analysis in the Aviation Instrument-Making Industry Nauchno-tekhnicheskii vestnik informatsionnykh tekhnologii, mekhaniki i optiki 2 1–17

[2] Belousov Yu A 2004 Fault–Tolerant Onboard Computing Systems. Classification and Performance Evaluation Aviakosmicheskoe priborostroenie 3 17–24

[3] Kolesov N V, Tolmacheva M V and Yukhta P V 2014 Real-Time Systems. Planning, Analysis, Diagnostics (St. Petersburg: Concern CSRI Elektropribor)

[4] Andreev Yu N 1976 Upravlenie konechnomernymi lineinymi ob'ektami (Moscow: Nauka - Control of Finite-Dimensional Linear Plants)
[5] Shumskii A E and Zhirabok A N 2009 Metody i algoritmy diagnostirovaniya i otkazoustoichivogo upravleniya dinamicheskimi sistemami (Vladivostok: DVGTU – Methods and Algorithms for Diagnosis and Fault-Tolerant Control of Dynamic Systems)

[6] Shtessel Y, Edwards C and Fridman L et al. 2014 Sliding Mode Control and Observation (New York: Springer New York)

[7] Wang W and Gao Z 2003 A Comparison Study of Advanced State Observer Design Techniques American Control Conference 6 4754-4759.

[8] Efimov D V and Raisi T 2016 Design of Interval Observers for Uncertain Dynamical Systems Automation and Remote Control 77(2) 191-225.

[9] Mazenc F and Bernard O 2011 Interval Observers for Linear Time-Invariant Systems with Disturbances Automatica 47(1) 140–147

[10] Kieffer M and Walter E 2004 Guaranteed Nonlinear State Estimator for Cooperative Systems Numerical Algorithms 37 187–198

[11] Kolesov N V, Gruzlikov A M and Lukoyanov E V 2017 Using Fuzzy Interacting Observers for Fault Diagnosis in Systems with Parametric Uncertainty Procedia Computer Science 103 499–504

[12] Gruzlikov A M, Kolesov N V and Lukoyanov E V 2018 Test-based diagnosis of faults in data exchange addressing in computer systems using parallel model Journal of Computer and Systems Sciences International 57(3) 420-433

[13] Gruzlikov A M, Kolesov N V, Lukoyanov E V and Tolmacheva M V 2018 Test-based diagnosis of distributed computer system using a time-varying model IFAC PapersOnLine 51(24) 1075–1082

[14] Burdonov I B, Kosachev A S and Kulyamin V V 2008 Teoriya sootvetstviya dlya sistem s blokirovkami i razrusheniem (Moscow: Nauka – Conformance Theory of the Systems with Refused Inputs and Forbidden Actions)

[15] Burdonov I B, Kosachev A S and Kulyamin V V 2000 Application of Finite Automatons for Program Testing Programming and Computer Software 26(2) 61-73

[16] Rivkin B S 2019 Russia’s first integrated navigation systems for commercial vessels Gyroscopy and Navigation 10(1) 35-40

[17] Mikrin E A, Mikhailov M V, Orlovskii I V, Rozhkov S N and Krasnopol'skii I A 2019 Satellite navigation of lunar orbiting spacecraft and objects on the lunar surface Gyroscopy and Navigation 27(1) 22-31

[18] Mikrin E A, Mikhailov M V, Orlovskii I V, Rozhkov S N, Semenov A S and Krasnopol'skii I A 2019 Circumlunar spacecraft navigation using the measurements from global navigation satellite systems GLONASS, GPS, GALILEO and BEIDOU Gyroscopy and Navigation 10(4) 187-195

[19] Gruzlikov A M and Kolesov N V 2016 Discrete-Event Diagnostic Model for a Distributed Computational System. Independent Chains Automation and Remote Control 10 1805-1817

[20] Cassandras C G and Lafortune S 2008 Introduction to discrete event systems (Dordrecht: Kluwer Academic Publishers)

[21] Zaytoon J and Lafortune S 2013 Overview of Fault Diagnosis Methods for Discrete Event Systems Annual Reviews in Control 37 308–320