ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2026, Vol. 87, No. 10, pp. 926–939.
© The Author(s), 2026 published by Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, 2026.
Russian Text © The Author(s), 2026, published in Avtomatika i Telemekhanika, 2026, No. 10, pp. 84–103.

NONLINEAR SYSTEMS

An Integrated Approach to Fault Accommodation in Fuzzy Discrete-Event Systems

A.E. Shumsky[*,**,a] **and A.N. Zhirabok**[*,**,b]

[*] *Far Eastern Federal University, Vladivostok, Russia*
[**] *Institute of Marine Technology Problems, Far Eastern Branch, Russian Academy of Sciences, Vladivostok, Russia*
e-mail: [a] *alexeyshumsky@yandex.ru,* [b] *zhirabok@mail.ru*

Received July 17, 2025
Revised January 12, 2026
Accepted January 23, 2026

Abstract—This paper considers the problem of fault accommodation in complex safety-critical discrete-event systems described by the fuzzy finite state automaton model. The problem is solved by forming control that ensures complete decoupling from fault-induced effects. The fault-tolerant control scheme contains several channels activated depending on the diagnosed fault. Each channel includes two main blocks: 1) an auxiliary deterministic finite state automaton (observer) estimating the fuzzy vector of possibilities of system transitions to various states and 2) a converter forming the input of the controlled system based on a new external input and the observer state. An integrated approach (the pair algebra of fuzzy coverings) is proposed to describe the blocks and the new input. A distinctive feature of this approach is the joint use of mathematical constructs of the pair algebra of coverings (system of sets) and fuzzy logic.

Keywords: fault-tolerant control, fault accommodation, discrete-event systems, fuzzy finite state automata, observers, pair algebra of fuzzy coverings, optimization approach

DOI: 10.7868/S1608303226100044

1. INTRODUCTION

The fault-tolerance requirement imposed on controlled safety-critical systems assumes the ability of such systems to adapt their behavior to faults arising during operation. Following the classification of adaptive systems into self-tuning and self-organizing systems, one distinguishes between two fundamentally different approaches to ensuring fault tolerance. The first approach (self-tuning or fault accommodation) is to form a fault-tolerant control law under which a faulty system will continue operation while preserving its most important performance indices within admissible limits. The second approach (self-organization) is to introduce structural changes into the system and its reconfiguration in order to remove failed elements and use reserve ones.

In this paper, we address the fault accommodation problem. In the monograph [1], it was solved using an original approach with the following stages: 1) fault detection, 2) fault isolation, 3) fault identification (determination of the fault value), 4) construction of a model of the faulty system, and, finally, 5) fault-tolerant control design based on the model. The approach to accommodation proposed therein involves mathematical models of controlled systems in the form of linear ordinary differential equations. Stages 1–3 are implemented using well-developed fault diagnosis methods for linear dynamic systems, and an appropriate control law (stage 5) is constructed by optimal control methods.

The efficiency of fault accommodation is primarily determined by the speed of forming the fault-tolerant control law. Therefore, the possibility of reducing the time costs of implementing each accommodation stage, or even excluding some stages, is crucial for improving accommodation. The second factor impacting the efficiency of accommodation is the accuracy of the controlled system model used to form the fault-tolerant control law. Consequently, it is necessary to apply nonlinear dynamic models, which most accurately describe the behavior of real systems, and robust methods for fault diagnosis and fault-tolerant control, which reduce the impact of uncontrolled factors (exogenous disturbances and model errors).

Several works, e.g., see [2, 3], are remarkable for the use of adaptive sliding mode observers for simultaneously estimating the state and parameters of the system and fault-tolerant control design methods based on sliding modes. Such a framework provides acceptable time costs for implementing the accommodation stages.

For controlled systems modeled by nonlinear ordinary differential equations, an approach without stages 3 and 4 was proposed in [4]. Its peculiarity is the use of a control law ensuring complete decoupling from the effects caused by faults. This approach was later extended to controlled systems described by discrete-time models in the form of nonlinear difference equations [5].

In addition to the above classes of continuous- or discrete-time controlled systems described by linear or nonlinear differential or difference equations, there exists an important class of so-called discrete-event systems (DESs) [6, 7]. These include not only systems that naturally fall into the class (e.g., digital information processing and control systems). Many complex systems conventionally classified as continuous, such as physical, technical (including human–machine ones), and socioeconomic systems, can be treated as DESs at the upper level of their hierarchy.

DESs possess the following distinctive features:

1. DESs are discrete in time and in the values of state variables.

2. The state space of a DES is finite; states may have different semantic content, e.g., “the DES is idle,” “the DES operates in a certain mode” (possibly caused by a system fault), “the operation of the DES is completed,” etc.

3. The operation of a DES is determined by events that may be consequences of certain external actions (inputs or commands) and faults, e.g., “start operation,” “change the operating mode of the fault-free DES,” “correct the control law when a certain fault is detected in the DES,” “terminate the operation of the DES,” etc.

4. As a rule, a DES behaves randomly, which is associated with the possibility of realizing different transitions from one state to another initiated by the same event.

The following mathematical models are commonly employed to describe DESs for subsequent fault diagnosis and fault-tolerant control design [8]:

1) deterministic finite state automata,

2) probabilistic finite state automata and Markov chains, and

3) Petri nets.

This paper focuses on the use of finite state automaton (FSA) models. By assumption, the inputs and outputs of an FSA are formed as observable events (external and internal, respectively), while the occurrence of a DES fault is treated as a directly unobservable internal event. A solution of the fault accommodation problem in a DES based on the deterministic FSA (DFSA) model and the complete decoupling principle was presented in [9]. This solution involves the mathematical apparatus of the pair algebra of partitions [10].

Whenever the DFSA model of a DES is unavailable, along with the probabilistic FSA model, models of nondeterministic and fuzzy FSAs (NFSA and FFSA, respectively) can also be applied. Compared with their probabilistic counterparts, nondeterministic and fuzzy models reduce the

volume of required computations, leading to lower time costs for fault accommodation. NFSA and FFSA models reflect the situation of transitions to different states from a certain fixed state and a certain fixed input of the automaton. Note that in the NFSA case, no priority can be assigned to the possibility of realizing a particular transition. In contrast, in the FFSA case, additional (e.g., statistical) information, expert estimates, learning results, etc., can be utilized to obtain the degree of confidence in the realization of each possible transition.

The aim of this paper is to extend the results of [9] to the case of FFSA models of DESs. The problem under consideration is to develop fault-tolerant control design methods oriented toward FFSA models of DESs. To solve this problem, we employ mathematical constructs of the pair algebra of coverings [10], fuzzy logic [11], and also the concepts of a fuzzy automaton [12] and its determinizer [13].

2. PROBLEM STATEMENT

Let the fault-free operation of a DES be described by the FFSA model

$$A = (U, X, Y, \delta, \lambda, x(0)), \tag{1}$$

where $U = \{u_1, \ldots, u_m\}$, $X = \{x_1, \ldots, x_n\}$, and $Y = \{y_1, \ldots, y_l\}$ are finite sets of inputs, states, and outputs, respectively; $x(0) \in X$ is a known initial state; $\delta : X \times U \to \mu(X)$ and $\lambda : X \to \mu(Y)$ are fuzzy transition and output functions, respectively; $\mu(X) \in \{\mu(x_i) \in [0,1], 1 \leqslant i \leqslant n\}$ and $\mu(Y) \in \{\mu(y_j) \in [0,1], 1 \leqslant j \leqslant l\}$ are fuzzy sets. For the fuzzy transition function δ, we will adopt a matrix representation in which transitions performed under an input $u_k \in U$ are described by a matrix M^k of dimensions $n \times n$: its element $M^k_{i,j} \in [0,1]$ characterizes the degree of confidence that the automaton A will pass from a state $x_i \in X$ to a state $x_j \in X$ under an input u_k. Let $S(U)$ denote the set of all matrices M^k, $1 \leqslant k \leqslant m$. To specify the fuzzy output function λ, we will use a matrix L of dimensions $l \times n$: its element $L_{i,j}$ determines the degree of confidence that the output $y_i \in Y$ will be formed by the automaton A in a state $x_j \in X$. (Note that the same output can be formed in different DES states.) Let $S(Y)$ denote the set of all rows L_i, $1 \leqslant i \leqslant l$, of the matrix L.

Model (1) corresponds to a special case of an FFSA with deterministic inputs and fuzzy transition and output functions [15]. If it is difficult to justify the degree of confidence in the possibility of transitions, one can proceed by setting $M^k_{i,j} = 1$ for a correct transition and $M^k_{i,j} = 0$ for an incorrect transition. Thereby, one will pass from the FFSA model of the DES to its NFSA model.

Let $X_{i,k}$, $X_{i,k} \subseteq X$, be the set of all states reachable from a state $x_i \in X$ under an input $u_k \in U$. Suppose that a fault f_s, $1 \leqslant s \leqslant N$, in the DES (1) can be represented by a distortion of the transition function δ such that under an input $u_k \in U$, an incorrect transition from a state x_i to a state $x_t \notin X_{i,k}$ is realized instead of an correct one to a state $x_j \in X_{i,k}$. We will write this fact as follows: $f_s : (x_j \to x_t)_{i,k}$. In this case, the matrix M^k_s of the DES with the fault f_s is obtained from the matrix M^k of the fault-free DES by changing the value of the element $M^k_{i,t}$ from 0 to 1. The analysis below is based on the hypothesis of single faults from a given list $F = \{f_1, \ldots, f_N\}$.

Consider the sequence and content of actions necessary for forming a control law that ensures complete decoupling from the effects caused by a fault f_s. We introduce the FFSA model of the DES with a fault f_s :

$$A_s = (U, X, Y, \delta_s, \lambda, x(0)). \tag{2}$$

The transition function δ_s of model (2) can be obtained by supplementing the set of transitions determined by the function δ of model (1) with a transition initiated by a fault $f_s \in F$ through changing the matrix M^k in the manner described above. Owing to the features of constructing the matrix M^k, model (2) reflects the possibility of a correct transition during the fault-free operation

of the DES and also the possibility of an incorrect transition to a state caused by a DES fault. Hence, we can speak of a certain correspondence between the behavior of model (2) and that of both the fault-free DES and the DES with a fault.

Following [9], let the input u ensuring accommodation to a fault f_s be formed in accordance with the law

$$u = g(x_0, y, u_*), \tag{3}$$

where g is a mapping to be determined; u_* is a new input; x_0 is the state of the deterministic FSA

$$A_0 = (X_0, U \times Y, \delta_0, x_0(0)), \tag{4}$$

whose transition function δ_0 is defined by the mapping $X_0 \times U \times Y \to X_0$ and is found by determinizing the automaton A_s; $x_0(0) \in X_0$ is the initial state of the automaton A_0; the output function of the automaton A_0 is not considered an independent object. Note that the description of the deterministic FSA A_0, hereinafter called the observer, does not depend on the (non-)occurrence of a DES fault f_s.

To find the control law ensuring accommodation of the DES (1) to a fault $f_s \in F$, it is necessary to solve the following problems sequentially:

1) construct model (2) of the DES with a fault f_s;

2) find the observer (4) by determinizing the automaton (2);

3) find the mapping g and form the control law (3).

These problems define the structure of the paper. In Section 3, we present an approach to determinizing an FFSA by fuzzy logic methods and reveal its drawback. To provide theoretical grounds for an approach that eliminates the drawback, we elaborate the algebra of fuzzy coverings (Section 4) and propose an algorithm for constructing an observer (Section 5). The issues of fault-tolerant control design are discussed in Section 6. An illustrative example is given in Section 7. The outcomes of this research are summarized in the Conclusions (Section 8).

3. DETERMINIZATION OF THE FUZZY FINITE STATE AUTOMATON A_s

Recall several operations of fuzzy logic that are important for the subsequent presentation. A fuzzy matrix $B = \{B_{ij}\}$ is a matrix with elements $B_{ij} \in [0, 1]$. Hence, the matrices from the set $S(U)$ introduced above are fuzzy matrices. Now, let B and C be fuzzy matrices of dimensions $a \times b$ and $b \times c$, respectively. The product of fuzzy matrices is defined as [13]

$$(BC)_{ij} = \max\limits_{ij} \min(B_{ih} C_{hj}) = \max(\min(B_{i1} C_{1j}), \ldots, \min(B_{ib} C_{bj}),$$

where $1 \leqslant i \leqslant a$, $1 \leqslant h \leqslant b$, and $1 \leqslant j \leqslant c$. This formula generalizes the well-known matrix multiplication one [16, p. 24]; it is obtained by replacing the product (sum) of matrix elements with their minimum (maximum, respectively). Here is a simple example.

Example 1. Consider matrices of the form

$$B = \begin{pmatrix} 0.1 & 0.9 \\ 0.7 & 0.2 \end{pmatrix}, \quad C = \begin{pmatrix} 0.6 & 0.3 \\ 0.4 & 0.8 \end{pmatrix}.$$

Straightforward calculations yield

$$\begin{aligned} (BC)_{11} &= \max(\min(0.1\ 0.6), \min(0.9\ 0.4)) = 0.4, \\ (BC)_{12} &= \max(\min(0.1\ 0.3), \min(0.9\ 0.8)) = 0.8, \\ (BC)_{21} &= (\min(0.7\ 0.6), \min(0.9\ 0.4)) = 0.6, \\ (BC)_{22} &= \max(\min(0.7\ 0.3), \min(0.2\ 0.8)) = 0.3. \end{aligned}$$

As a result, the product of fuzzy matrices is

$$BC = \begin{pmatrix} 0.4 & 0.8 \\ 0.6 & 0.3 \end{pmatrix}.$$

We denote by E the set of fuzzy vectors of dimensions $1 \times n$; their components can take values in the interval $[0, 1]$. Let $A' = (U, X, \delta)$ be a fuzzy semiautomaton (i.e., an automaton A without the output function) and $S(U)$ be the set of fuzzy transition matrices of the automaton. Suppose that the signature (the set of admissible operations) Σ includes all operations involving multiplication of a fuzzy vector of dimensions $1 \times n$ on the right by a matrix from the set $S(U)$. Also, let $E^0 \subseteq E$ be the set of all unit vectors of dimensions $1 \times n$, called generating states of the determinizer. The closure $[E]_\Sigma$ of the set E with respect to the signature Σ is the set of all vectors (including those of E) that can be obtained by applying operations from Σ to vectors from E. We propose the following algorithm.

Algorithm 1 (construction of the closure).

1. Set $i = 0$.
2. Find the set of vectors $E^{i+1} = E^i \cup [E^i]_\Sigma$.
3. If $E^{i+1} = E^i$, set $[E]_\Sigma = E^i$. End.
4. Assign $i := i + 1$ and return to step 2.

According to [13], the determinizer $D(A')$ of the FFSA A' is a DFSA described by the triple

$$D(A') = (S(X), U, \delta_D), \tag{5}$$

where $S(X) = [E]_{S(U)}$ and $\delta_D : S(X) \times S(U) \to S(X)$ is the determinizer's transition function defined as

$$\delta_D(\mu_i, u_k) = \mu_i M^k, \quad \mu_i \in S(X), \quad M^k \in S(U). \tag{6}$$

Thus, each state μ_i of the determinizer is associated with a certain fuzzy vector of membership grades characterizing the possibility of reaching certain states of the controlled DES during its operation.

In [14], the maximum number of states of the determinizer was estimated as

$$\sharp S(X) \leqslant q^n - 1, \tag{7}$$

where the symbol $\sharp$ denotes cardinality; q is the number of different values of elements of fuzzy matrices from the set $S(U)$; as before, n is the number of states of the FFSA.

According to the relation (7), the number of states of the determinizer can substantially exceed that of the original automaton and grows exponentially with the latter. This fact has minimal impact on fault diagnosis since such problems are solved in real time without requiring the complete description of the determinizer [14]. However, in the case of fault accommodation, additional computations are needed to find the mapping g and choose the control u_* based on the determinizer's description. The volume and feasibility of computations will be directly related to the determinizer's dimension. As a consequence, there may exist a restriction on the dimension of the original system model. To overcome this restriction, we propose to design the observer A_0 with another set of states, i.e., to replace the closure E with a certain set not exceeding, by the dimension, the original model (1), while preserving its closedness with respect to operations from the signature Σ. To find such a set, the pair algebra of fuzzy coverings will be introduced below.

Another difference between the observer and the determinizer is that the former's transition function also depends on the DES output. Therefore, it is possible to refine the membership grades (the components of fuzzy vectors associated with x_0) considering the additional information contained in the DES output. This feature explains the term "observer" used for the automaton A_0.

4. THE PAIR ALGEBRA OF FUZZY COVERINGS

In [17], constructs of the pair algebra of coverings were introduced for solving the determinization problem of NFSAs. Below, these constructs are extended to fuzzy sets to propose an approach to determinizing FFSAs considering the information contained in the fuzzy model of DESs.

Recall that a covering π of a set X is a set of its subsets (blocks of the covering) $\pi = \{B_{\pi,1}, \ldots, B_{\pi,\nu}\}$ such that

$$B_{\pi,k} \subseteq X, \qquad \bigcup_{k=1}^{\nu} B_{\pi,k} = X, \qquad B_{\pi,k} \subseteq B_{\pi,j} \Rightarrow B_{\pi,k} = B_{\pi,j}.$$

Note that under the additional condition

$$B_{\pi,k} \cap B_{\pi,j} = \emptyset, \quad k \neq j,$$

the above constructs define a partition of X. The set of all coverings of a set X is denoted by Π_X.

Assume that the blocks of the coverings considered below are fuzzy sets. It is convenient to specify such blocks by a row vector of membership functions. In this case, the notation $\mu_{B_{\pi,j}} = (0\ \ 0.1\ \ 0.7\ \ 0\ \ 0\ \ 0.5)$ means that (a) the cardinality of X is 6, (b) the jth block of the covering π contains the states x_2, x_3, and x_6, and (c) the nonzero membership grades of these states for the jth block of the covering are $\mu_{B_{\pi,j}}(x_2) = 0.1$, $\mu_{B_{\pi,j}}(x_3) = 0.7$, and $\mu_{B_{\pi,j}}(x_6) = 0.5$.

To perform operations on coverings, it is necessary to find the intersections and unions of their blocks. Let π and σ be certain coverings of a set X. In accordance with the rules of operations on fuzzy sets [14], we write

$$\mu_{B_{\pi,j} \cap B_{\sigma,k}}(x_i) = \min(\mu_{B_{\pi,j}}(x_i), \mu_{B_{\sigma,k}}(x_i)),$$
$$\mu_{B_{\pi,j} \cup B_{\sigma,k}}(x_i) = \max(\mu_{B_{\pi,j}}(x_i), \mu_{B_{\sigma,k}}(x_i)).$$

A covering π is said to be less than or equal to a covering σ, $\pi \leqslant \sigma$, if, for each block $B_{\pi,j}$ of π, there always exists a block $B_{\sigma,k}$ of σ such that $B_{\pi,j} \subseteq B_{\sigma,k}$. In this case, $B_{\pi,j} \cup B_{\sigma,k} = B_{\sigma,k}$, and hence the inclusion $B_{\pi,j} \subseteq B_{\sigma,k}$ requires the fuzzy set $B_{\sigma,k}$ to dominate the fuzzy set $B_{\pi,j}$, i.e.,

$$\mu_{B_{\pi,j}}(x_i) \leqslant \mu_{B_{\sigma,k}}(x_i), \quad 1 \leqslant i \leqslant n. \tag{8}$$

There are two special coverings, denoted by $\mathbf{0}$ and $\mathbf{1}$, representing partitions such that each block of $\mathbf{0}$ contains only one element of a set X, while $\mathbf{1}$ includes only one block containing all elements of X. For an arbitrary covering π of a set X, we have

$$\mathbf{0} \leqslant \pi \leqslant \mathbf{1}.$$

As is known [10], the set of all coverings of a set X with a partial order relation forms a lattice. Consequently, for each pair of coverings π and σ, one can find the two coverings $\inf(\pi, \sigma)$ and $\sup(\pi, \sigma)$. These coverings are conventionally denoted by $\pi \times \sigma$ and $\pi + \sigma$, respectively (the product and sum of coverings). To calculate them, we use the rules formulated in [17].

1. Each block of the covering $\pi \times \sigma$ is the intersection of certain blocks of the coverings π and σ.
2. Each block of the covering $\pi + \sigma$ is the union of intersecting blocks of the coverings π and σ.

We emphasize a peculiarity that may arise during the operation $\times$. Let the zero partition be the result of this operation in accordance with the rules of traditional algebra of coverings [10]. However, with formal application of the above method considering membership functions, blocks of the zero partition may appear for which the values of the membership functions of states are strictly less than 1. This indicates a singularity in the process of transforming the FFSA model of

the system (e.g., during its determinization). If necessary, a possible recipe here is to pass (fully or partially) to the NFSA model of the system by replacing the nonzero values of the corresponding membership functions with 1.

Example 2. Consider $X = \{x_1, x_2, x_3, x_4, x_5, x_6\}$ and coverings $\pi = \{(x_1); (x_3); (x_2, x_4); (x_2, x_5, x_6)\}$, $\sigma = \{(x_1); (x_2, x_5); (x_3, x_4); (x_4, x_6)\}$ with the membership functions of states for the corresponding blocks $\mu_{B_{\pi,4}}(x_2) = 0.5$ and $\mu_{B_{\sigma,3}}(x_4) = 0.3$; the other membership functions are equal to 1. By applying the rule for the intersection of fuzzy sets to the blocks of the coverings π and σ, we obtain the membership functions $\mu_{B_{\pi,1}\cap B_{\sigma,1}}(x_1) = \mu_{B_{\pi,2}\cap B_{\sigma,3}}(x_3) = \mu_{B_{\pi,3}\cap B_{\sigma,2}}(x_2) = \mu_{B_{\pi,3}\cap B_{\sigma,4}}(x_4) = \mu_{B_{\pi,4}\cap B_{\sigma,4}}(x_6) = 1$, $\mu_{B_{\pi,4}\cap B_{\sigma,2}}(x_5) = 0.5$, and $\mu_{B_{\pi,3}\cap B_{\sigma,3}}(x_4) = 0.3$. The membership functions of states for the empty sets obtained by intersecting blocks of the partitions π and σ are zero. Note that $B_{\pi,3\cap\sigma,2} \subseteq B_{\pi,4\cap\sigma,2}$. Following the definition of a covering, it is necessary to replace both of these blocks by one block $B_{\pi,3\cap\sigma,2} \cup B_{\pi,4\cap\sigma,2}$ dominating them. Moreover, the fuzzy set $B_{\pi,3} \cap B_{\sigma,4}$ dominates the fuzzy set $B_{\pi,3} \cap B_{\sigma,3}$. As a result, we have arrived at the partition $\pi \times \sigma = \{(x_1); (x_3); (x_2, x_5); (x_4); (x_6)\}$.

Now, let us find the sum of these coverings, $\pi+\sigma$. By uniting the intersecting blocks of coverings, we obtain

$$\begin{gathered}\mu_{B_{\pi,1}\cup B_{\sigma,1}}(x_1) = \mu_{B_{\pi,2}\cup B_{\sigma,3}}(x_3) = \mu_{B_{\pi,2}\cup B_{\sigma,3}}(x_4) = \mu_{B_{\pi,4}\cup B_{\sigma,2}}(x_2)\\ = \mu_{B_{\pi,3}\cup B_{\sigma,2}}(x_4) = \mu_{B_{\pi,3}\cup B_{\sigma,2}}(x_5) = \mu_{B_{\pi,3}\cup B_{\sigma,3}}(x_3) = \mu_{B_{\pi,3}\cup B_{\sigma,3}}(x_4)\\ = \mu_{B_{\pi,4}\cup B_{\sigma,2}}(x_2) = \mu_{B_{\pi,4}\cup B_{\sigma,2}}(x_5) = \mu_{B_{\pi,4}\cup B_{\sigma,2}}(x_6) = \mu_{B_{\pi,4}\cup B_{\sigma,3}}(x_2)\\ = \mu_{B_{\pi,4}\cup B_{\sigma,3}}(x_3) = \mu_{B_{\pi,4}\cup B_{\sigma,3}}(x_4) = \mu_{B_{\pi,4}\cup B_{\sigma,3}}(x_5) = \mu_{B_{\pi,4}\cup B_{\sigma,3}}(x_6)\\ = \mu_{B_{\pi,2}\cup B_{\sigma,4}}(x_2) = \mu_{B_{\pi,4}\cup B_{\sigma,4}}(x_5) = \mu_{B_{\pi,4}\cup B_{\sigma,4}}(x_6) = 1,\\ \mu_{B_{\pi,3}\cup B_{\sigma,3}}(x_2) = 0.5, \quad \mu_{B_{\pi,4}\cup B_{\sigma,4}}(x_3) = 0.3.\end{gathered}$$

When constructing the resulting covering, we take into account that $B_{\pi,3} \cup B_{\sigma,3}$ dominates $B_{\pi,2} \cup B_{\sigma,3}$ and also $B_{\pi,3} \cup B_{\sigma,2} \subseteq B_{\pi,4} \cup B_{\sigma,4}$, $B_{\pi,3} \cup B_{\sigma,4} \subseteq B_{\pi,4} \cup B_{\sigma,4}$, and $B_{\pi,4} \cup B_{\sigma,2} \subseteq B_{\pi,4} \cup B_{\sigma,4}$. As a consequence, the dominating fuzzy sets $B_{\pi,3} \cup B_{\sigma,3}$ and $B_{\pi,3} \cup B_{\pi,4} \cup B_{\sigma,2} \cup B_{\sigma,4}$ are chosen as the independent blocks of the covering $\pi + \sigma$. The final covering is $\pi + \sigma = \{(x_1); (x_2, x_3, x_4); (x_2, x_4, x_5, x_6)\}$, where the second block represents a fuzzy set since $\mu B_{\pi+\sigma,2}(x_2) = 0.5$.

Let π and σ be certain coverings of a set X. For the convenience of further presentation, we give a concise description only for the main constructs of the pair algebra of coverings (used in this paper), i.e., the binary relation Δ, the concept of pair coverings, and the operator $\mathbf{m}$. More complete information on this issue can be found, e.g., in [10].

Coverings π and σ are said to satisfy the relation Δ, $(\pi, \sigma) \in \Delta$, and form a pair if

$$x \equiv x'(\pi) \Rightarrow \delta(x, u_k) \equiv \delta(x', u_k)(\sigma) \quad \forall u_k \in U. \tag{9}$$

Let $x, x' \in B_{\pi,j}$. According to [17], for each $u_k \in U$, we can write $\delta(x, u_k), \delta(x', u_k) \in B^k_{\sigma,j}$, where the vector of membership functions $\mu_{B^k_{\sigma,j}}$ is defined by

$$\mu_{B^k_{\sigma,j}} = \mu_{B(\pi,j)} M^k, \quad M^k \in M(U). \tag{10}$$

For a given covering π, there may exist several coverings satisfying the relation $(\pi, \sigma) \in \Delta$. In particular, $(\pi, \mathbf{1}) \in \Delta$ is valid for any covering π. We introduce the operator $\mathbf{m}$ that bijectively associates the covering $\mathbf{m}(\pi)$ with a covering π as follows:

$$(\pi, \mathbf{m}(\pi)) \in \Delta, \quad (\pi, \sigma) \in \Delta \Rightarrow \mathbf{m}(\pi) \leqslant \sigma.$$

The operator $\mathbf{m}(\pi)$ is computed by the following algorithm.

Algorithm 2 (computation of the operator $\mathbf{m}(\pi)$).

1. For each $u_k \in U$ and each j, $1 \leqslant j \leqslant \nu$, where ν is the number of blocks of a covering π, find the subset $B^k_{\sigma,j}$ using (10).

2. Write the set of non-empty subsets $Q = \{B^k_{\sigma,j}, 1 \leqslant k \leqslant m, 1 \leqslant j \leqslant \nu\}$.

3. Form the covering $\mathbf{m}(\pi)$ from Q by uniting those subsets for which one is contained in another. End.

Example 3. Consider an FFSA described by the matrices

$$M^1 = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad M^2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.7 & 0.3 & 0 \\ 0 & 0.4 & 0.6 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad M^3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0.9 & 0 & 0 & 0.1 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix},$$

and set $\pi = \mathbf{0}$. It is required to compute $\sigma = \mathbf{m}(\pi)$. When executing step 1 of Algorithm 2, with the first block $\mu_{B_{\pi,1}} = (1\ 0\ 0\ 0\ 0)$ of the covering π taken as a basis, we obtain the vector membership function $\mu_{B^1_{\sigma,1}} = \mu_{B_{\pi,1}} M^1 = (0\,1\,0\,0\,0)$ corresponding to the subset $B^1_{\sigma,1} = \{x_2\}$. Next, the vector membership functions $\mu_{B^2_{\sigma,1}} = \mu_{B_{\pi,1}} M^2 = (0\ 0\ 0\ 0\ 0)$ and $\mu_{B^3_{\sigma,1}} = \mu_{B_{\pi,1}} M^3 = (0\ 0\ 0\ 0\ 0)$, corresponding to the empty subsets $B^2_{\sigma,1}$ and $B^3_{\sigma,1}$, are found. For the remaining blocks of the covering π, similar computations yield non-empty fuzzy subsets with the membership functions $\mu_{B^3_{\sigma,1}} = (0\ 0\ 1\ 0\ 0)$, $\mu_{B^2_{\sigma,2}} = (0\ 0\ 0.7\ 0.3\ \ 0)$, $\mu_{B^3_{\sigma,2}} = (1\ 0.7\ 0\ 0\ 0)$, $\mu_{B^2_{\sigma,3}} = (0\ 0.4\ 0.6\ 0\ 0)$, $\mu_{B^3_{\sigma,3}} = (0\ 0\ 0.9\ 0\ 0.1)$, $\mu_{B^3_{\sigma,4}} = (0\ 0\ 0\ 0\ 1)$, and $\mu_{B^3_{\sigma,5}} = (1\ 0\ 0\ 0\ 0)$. In accordance with step 2 of Algorithm 2, $Q = \{B^3_{\sigma,5}; B^1_{\sigma,1}; B^2_{\sigma,3}; B^2_{\sigma,2}; B^3_{\sigma,3}; B^3_{\sigma,4}\} = \{(x_1); (x_2); (x_2, x_3); (x_3, x_4); (x_5)\}$. For step 3 of the algorithm, note that $B^1_{\sigma,1} \subseteq B^2_{\sigma,3}$ and $B^3_{\sigma,4} \subseteq B^3_{\sigma,3}$. We replace the above fuzzy sets with the dominating fuzzy ones $B^* = B^1_{\sigma,1} \cup B^2_{\sigma,3}$ and $B^{**} = B^3_{\sigma,4} \cup B^3_{\sigma,3}$ with the membership functions $\mu_{B^*} = (0\ 1\ 0.6\ 0\ 0)$ and $\mu_{B^{**}} = (0\ 0\ 0.9\ 0\ 0.1)$, respectively. Finally, we arrive at the fuzzy covering $\sigma = \{B^3_{\sigma,5}; B*; B^2_{\sigma,3}; B^{**}\} = \{(x_1); (x_2, x_3); (x_3, x_4); (x_3, x_5)\}$ with the corresponding membership functions.

Note that, in contrast to DFSAs, the relation $\mathbf{m}(\mathbf{0}) \neq \mathbf{0}$ always holds for NFSAs or FFSAs.

5. DESIGN OF THE OBSERVER A_0

For an arbitrary covering φ, by analogy with the constructs of the algebra of functions for continuous-time dynamic systems [4], we introduce the concepts of δ-invariance,

$$(\varphi, \varphi) \in \Delta,$$

and (λ, δ)-invariance,

$$(\pi_\lambda \times \varphi, \varphi) \in \Delta, \tag{11}$$

where the blocks of a fuzzy covering π_λ are specified by the rows of the matrix L. Obviously, δ-invariance is a special case of (λ, δ)-invariance (for $\pi_\lambda = \mathbf{1}$). The latter concept will be used to design the observer A_0. It has the following informational meaning [10]. Assume that at some discrete time instant, there is complete information about the DES output or, equivalently, information about the current DES state up to some block of a covering π_λ, as well as additional information about the current state up to some block of a covering φ. Then the DES state at the next discrete time instant can surely be calculated up to some block of the covering φ.

A (λ, δ)-invariant covering such that

$$\varphi^0 \leqslant \varphi, \tag{12}$$

where φ^0 is a given covering, can be found using an algorithm similar to the one presented in [15]; by assumption, the operations and operators of the algebra of coverings are computed in accordance with the rules of fuzzy logic introduced above.

Algorithm 3 (computation of a (λ, δ)-invariant covering φ).

1. Set $i = 0$ and specify φ^0.
2. Compute the covering

$$\varphi^{i+1} = \varphi^i + \mathbf{m}(\pi_\lambda \times \varphi); \tag{13}$$

if $\varphi^{i+1} = \varphi^i$, then go to step 3; otherwise, assign $i := i + 1$ and repeat step 2.

3. Assign $\varphi^{i+1} := \varphi^i$. End.

The convergence of Algorithm 3 and the property of the resulting covering are established by the following result.

Theorem 1 (convergence theorem). *There exists a finite number $i = i_{\max}$ such that the covering $\varphi = \varphi^i = \varphi^{i+1} = \varphi^{i_{\max}}$ determined by* (13) *is the smallest one satisfying conditions* (11) *and* (12).

Proof. For the case of partitions, it was provided in [9]. When using the algebra of coverings, the proof remains the same [17].

We introduce the operation

$$< L_i, \mu_{B_(\varphi,j)} >= (\min(L_{i,1}, \mu_{B_{\varphi,j}}(x_1)), \ldots, \min(L_{i,n}, \mu_{B_{\varphi,j}}(x_n))). \tag{14}$$

The result of this operation defines the row vector of membership functions of DES states for the block of the covering $\pi_\lambda \times \varphi$ obtained by intersecting the ith and jth blocks of the coverings π_λ and φ, respectively. Then, in view of (10), we have

$$\mu_{B^k_{\pi_\lambda \times \varphi^i}} =< L_i, \mu_{B_{\varphi^i_j}} > M^k_s, \tag{15}$$

where $B^k_{\pi_\lambda \times \varphi^i}$ is the subset from step 2 of Algorithm 2 when computing the operator $\mathbf{m}(\pi_\lambda \times \varphi^i)$. The relationship (15) is convenient to use when computing the fuzzy covering φ^{i+1} by formula (13).

By setting $\varphi^0 = \mathbf{0}$, we apply Algorithm 2 to find the covering φ. For each state $x_{0,j}$ of the observer A_0, we assign a block of the covering φ by using a representation in the form of a unit vector of dimensions $1 \times n_0$ with the jth element equal to 1. Here, n_0, $n_0 \leqslant n$, is the number of blocks of the covering φ, and the set X_0 of the unit vectors has a dimension not exceeding that of the state set of the original automaton A.

The equality

$$< L_i, \mu_{B_{\varphi,j}} >= 0 \tag{16}$$

indicates that the ith DES output is incompatible with the jth state of the DFSA A_0.

We will specify the transition function δ_0 of the observer by the family of matrices $M^{k,i}_0$ ($1 \leqslant k \leqslant m$, $1 \leqslant i \leqslant l$), each corresponding to a certain combination of values of the DES inputs and outputs. Next, we find

$$\mu_{B^+_{\varphi},j} =< L_i, \mu_{B_{\varphi,j}} > M^k_s. \tag{17}$$

If the inclusion

$$B^+_{\varphi,j} \subseteq B_{\varphi,t} \tag{18}$$

holds, the DES states from the block $B_{\varphi,j}$ of the covering φ under the output y_i and input u_k are transferred to those belonging to the block $B_{\varphi,t}$. Hence, the matrices $M_0^{k,i}$ can be found by computing (17) and checking the inclusion (18) for various combinations of the DES inputs and outputs: $1 \leqslant k \leqslant m$, $1 \leqslant i \leqslant l$. Such a check can be performed using the following algorithm. (The variables j and t correspond to the states of the observer A_0 before and after the transition.)

Algorithm 4 (computation of the matrices $M_0^{k,i}$).

1. Set $k = 1$.
2. Set $i = 1$.
3. Set $M_0^{k,i} = 0_{n_0 \times n_0}$.
4. Set $j = 1,\ t = 1$.
5. If (16) holds, assign $j := j + 1,\ t := 1$, and repeat step 5.
6. Find the set $B^+_{\varphi,j}$ from the row vector of the membership functions computed by (17).
7. Check the inclusion (18). If it fails, go to step 9.
8. Write "1" for the element of the matrix $M_0^{k,i}(i,t)$ standing at the intersection of the jth row and tth column. Go to step 10.
9. If $t = n_0$, assign $t := t + 1$ and return to step 7.
10. If $j = n_0$, assign $j := j + 1,\ t := 1$ and return to step 5.
11. If $i = l$, assign $i := i + 1$ and return to step 3.
12. If $k = m$, assign $k := k + 1$ and return to step 2. End.

The algorithm assumes execution of several nested loops for elementary actions on matrices; the total number of loops does not exceed lmn_0^2, which characterizes its computational complexity. Steps 1–4 specify the initial values of the variables t, j, i, and k, which control the loop execution sequence of Algorithm 4. Steps 9–12 serve to correct and control the limit values of these variables, ensuring the convergence of the algorithm. Since all computations are off-line, the algorithm can be implemented for real systems without significant computational resources, on a standard PC. The main part of the algorithm (steps 5–8) is illustrated below.

Example 4. Let the DES dynamics during fault-free operation be described by the FFSA model from Example 3. In addition, let the output function of the DES be specified by a fuzzy matrix L. Suppose that a fault $f_1 : (x_5 \to x_1)_{4,3}$ is possible in the system. As a consequence, the matrix M^3 takes the form of M_1^3. Here are the newly introduced matrices for the DES with the fault:

$$L = \begin{pmatrix} 1 & 0.5 & 0 & 1 & 0 \\ 0 & 0.5 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}, \quad M_1^3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9 & 0 & 0.1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

(The matrices M^1 and M^2 retain their previous form.)

For the FFSA model under consideration, we apply Algorithm 3 to find the (λ, δ)-invariant covering for $\varphi^0 = 0$. The operator $\mathbf{m}$ will be computed in accordance with Algorithm 2. By setting $i = 0$ and using (13), we obtain

$$\varphi^1 = \mathbf{0} + \mathbf{m}(\lambda \times \mathbf{0}) = \mathbf{m}(\mathbf{0}) = \{B^{***}; B^{*}; B^{2}_{\sigma,2}; B^{**}\} = \{(x_1, x_5); (x_2, x_3); (x_3, x_4); (x_3, x_5)\},$$

where $\mu_{B^{***}} = (1\ 0\ 0\ 0\ 1)$. Continuing computations based on (13), one can show that $\varphi^2 = \varphi^1 + \mathbf{m}(\pi_\lambda \times \varphi^1) = \varphi^1$. Following step 3 of Algorithm 3, we set $\varphi = \varphi^1$.

For the sake of convenience, we begin the illustration of Algorithm 4 by presenting some intermediate results. The table below demonstrates the results of checking the inclusion (18) (steps 5–8 of the algorithm).

The results of Algorithm 4 (steps 1–7)

$B_{\varphi 1} \cap B_{\pi_\lambda 1}$	M^1	M^2	M^3	$B_{\varphi 1} \cap B_{\pi_\lambda 2}$	M^1	M^2	M^3	$B_{\varphi 1} \cap B_{\pi_\lambda 3}$	M^1	M^2	M^3
$B^+_{\varphi 1} \subseteq$	$B_{\varphi 2}$	$\emptyset$	$\emptyset$	$B^+_{\varphi 1} \subseteq$	$\emptyset$	$\emptyset$	$B_{\varphi 1}$	$B^+_{\varphi 1} \subseteq$	$\emptyset$	$\emptyset$	$\emptyset$
$B_{\varphi 2} \cap B_{\pi_\lambda 1}$	M^1	M^2	M^3	$B_{\varphi 2} \cap B_{\pi_\lambda 2}$	M^1	M^2	M^3	$B_{\varphi 2} \cap B_{\pi_\lambda 3}$	M^1	M^2	M^3
$B^+_{\varphi 2} \subseteq$	$\emptyset$	$B_{\varphi 3}$	$\emptyset$	$B^+_{\varphi 2} \subseteq$	$\emptyset$	$B_{\varphi 3}$	$\emptyset$	$B^+_{\varphi 2} \subseteq$	$\emptyset$	$B_{\varphi 2}$	$B_{\varphi 4}$
$B_{\varphi 3} \cap B_{\pi_\lambda 1}$	M^1	M^2	M^3	$B_{\varphi 3} \cap B_{\pi_\lambda 2}$	M^1	M^2	M^3	$B_{\varphi 3} \cap B_{\pi_\lambda 3}$	M^1	M^2	M^3
$B^+_{\varphi 3} \subseteq$	$\emptyset$	$\emptyset$	$B_{\varphi 1}$	$B^+_{\varphi 3} \subseteq$	$\emptyset$	$\emptyset$	$\emptyset$	$B^+_{\varphi 3} \subseteq$	$\emptyset$	$B_{\varphi 2}$	$B_{\varphi 4}$
$B_{\varphi 4} \cap B_{\pi_\lambda 1}$	M^1	M^2	M^3	$B_{\varphi 4} \cap B_{\pi_\lambda 2}$	M^1	M^2	M^3	$B_{\varphi 4} \cap B_{\pi_\lambda 3}$	M^1	M^2	M^3
$B^+_{\varphi 4} \subseteq$	$\emptyset$	$\emptyset$	$\emptyset$	$B^+_{\varphi 4} \subseteq B_{\varphi 2}$	$\emptyset$	$B_{\varphi 2}$	$B_{\varphi 1}$	$B^+_{\varphi 4} \subseteq$	$\emptyset$	$B_{\varphi 2}$	$B_{\varphi 4}$

Horizontally, this table is split into four parts, each corresponding to a certain covering block $B_{\varphi,j}$ (the observer's state $x_{0,j}$), $1 \leqslant j \leqslant 4$, existing before the transition. Each part of the table contains two rows. The odd rows describe the arguments: combinations of the initial covering block $B_{\varphi,j}$ with the blocks of the covering π_λ (corresponding to the DES outputs) and with the matrices M^k_s used in the computations (corresponding to the DES inputs). The even rows present the blocks of the covering φ corresponding to the next states of the observer after the transition. Based on this table, we consider the process of finding the rows of matrices describing the dynamics of the observer A_0. Note that steps 1–4 of the algorithm yield $k = 1$, $i = 1$, $M_0^{1,1} = 0_{4\times 4}$, $j = 1$, and $t = 1$. Equality (16) fails, and (in accordance with step 5 of the algorithm) it is necessary to go to step 6. Computations of step 6 and a check of the inclusion (18) (step 7) lead to step 9. The subsequent return to step 7 and a check of the inclusion (18) confirm the latter's validity for $j = 1$ and $t = 2$. After executing step 8, the first row of the matrix $M_0^{1,1}$ will contain 1 in the second position. By continuing Algorithm 4 for the other values of the variable j, $2 \leqslant j \leqslant 4$, one will easily verify that the remaining rows of this matrix are 0. Thus, for a certain pair of inputs and outputs, the form of all rows of the matrix $M_0^{k,i}$ is completely defined by the table. For example, the elements of the third row of the matrix $M_0^{k,2}$ are 0 for all $k \in \{1, 2, 3\}$, while the elements of the fourth row of the matrices $M_0^{2,3}$ and $M_0^{3,3}$ contain 1 in the second and fourth positions, respectively.

6. FORMATION OF THE CONTROL LAW

Consider the following terminal control problem for a DES: it is required to transfer the system from an initial state $x(0) \in X$ to a certain final state x^r, $x^r \in X$. Note that the solution of this problem cannot be guaranteed based on a single reference trajectory due to an uncertain response of the DES to the same input actions. In other words, one can speak of controlling the system states only up to blocks of a certain covering of the set X. As a result, in practice, the existence of several correct trajectories may be required, with the possibility of transition between them depending on the DES response. The problem of forming correct alternative trajectories is closely related to the specifics of a particular system and will not be considered in detail here.

If a fault f_s is diagnosed in the DES, the mentioned covering is a covering φ [9]. Hence, the original terminal control problem is replaced by that of transferring the system state from a block B^0_φ of the covering φ, $x(0) \in B^0_\varphi$, to a block B^r_φ of the covering φ, $x^r \in B^r_\varphi$. Next, it is necessary to form the required trajectory in terms of blocks of the covering φ, representing a sequence of transitions: $B^0_\varphi \to B^1_\varphi \to \cdots \to B^r_\varphi$, where at least one state of the subsequent block is reachable from at least one state of the previous block. Note that the specified sequence will be realized regardless of the errors in DES transitions caused by the fault f_s since the resulting distortions of its state lie within the same block of the covering φ. Also, we emphasize that this sequence bijectively corresponds to a certain sequence of transitions of the observer $A_0 : x^0_0 \to x^1_0 \to \cdots \to x^r_0$.

Following [9], we will associate the value of the generated new input u_* with the current value of the transition function δ_0 of the observer or, equivalently, with the block $B_{\varphi,t}$ of the covering φ

whose states are reachable from the states of the block $B_{\varphi,j}$ (see (17) and (18)). For example, if the above chain of transitions contains a transition $B^i_\varphi \to B^{i+1}_\varphi$, $1 \leqslant i \leqslant r-1$, then the block B^i_φ here can be treated as the block $B_{\varphi,j}$, and B^{i+1}_φ as $B_{\varphi,t}$. Consequently, the new input u_* is completely defined by the desired next block of the covering φ in the chain of transitions.

Consider now the rule for forming the mapping g. This rule is based on analysis of the transition function δ_0 of the observer. The problem here is that for the same pair of combinations of the DES output y and the observer's state x^i_0 associated with the block B^i_φ of the covering φ, there may exist several inputs implementing the required transition of the DES to the state $x_* \in X$. To eliminate the arising ambiguity, we choose an input ensuring the highest degree of confidence in the transition of the system to the specified state x_* in accordance with the criterion

$$J = \max_u \mu_{B^{i+1}_\varphi}(x_*). \tag{19}$$

Example 5. We return to the DES with the fault f_1 from Examples 3 and 4. Let the terminal control problem consist in sequentially transferring this system from the initial state $x(0) = x_1 \in B_{\varphi,1}$ to the final state $x^r = x_3$ by realizing the chain of transitions $x_1 \to x_2 \to x_3$. We associate this chain with a sequence of blocks of the covering φ. Since $x_2 \in B_{\varphi,2}$, the only correct transition is ensured by the DES input u_1. This can be verified using the table as a basis for forming the mapping g. However, if the output y_3 is formed instead, then the transition is implemented to the state x_3 rather than to the state x_2. In other words, the originally specified trajectory (chain of DES transitions) is violated, and a transition to another correct trajectory is needed.

If another DES output is formed (not y_3), then the transition to the state x_2 can be ensured. In this case, according to the table, the input u_2 will implement the next transition to the block $B_{\varphi,3}$, containing the state x_3. The formation of the output y_3 will then confirm the successful transition.

Let us illustrate the optimization approach based on the criterion (19). Suppose now that $x_3 \in B_{\varphi,2}$, the DES output is y_3, and while realizing the alternative trajectory $x_1 \to x_3 \to x_3$, it is required to preserve the same state ($x_* = x_3$) with the greatest confidence. According to the table, the state can be preserved with the inputs u_2 or u_3. After a successful transition, we have $x_3 \in B_{\varphi,2}$ or $x_3 \in B_{\varphi,4}$, respectively. However, in the latter case, the value of the criterion J is greater than in the former: $J = \mu_{B_{\varphi,4}}(x_3) = 0.9$ vs. $J = \mu_{B_{\varphi,2}}(x_3) = 0.6$. Therefore, the choice u_3 is preferable.

7. AN ILLUSTRATIVE EXAMPLE

Consider the process of change management in an IT system [17, Table 1]. This process is one of the most important for IT systems: it is responsible for managing the life cycle of all changes and facilitates the implementation of useful changes with minimal interruption of IT services. The agents of the process are:

1) the initiator (an IT department representative who performs the initial processing, assignment, and control of changes);

2) the executor (an engineer who makes changes to configuration elements or coordinates the contractor's work on these changes);

3) the Advisory Change Committee (ACC, an advisory body that meets regularly to assess and plan changes);

4) the process manager (an IT department representative who controls the change management process and forms suggestions for its improvement).

In [14], the fuzzy finite state automaton model of a process was introduced, and a solution of the fault diagnosis problem of the process was proposed. Omitting the detailed description of the model itself, we present the list of the variables used. The following external events are considered to be model inputs: u_1 (work plan completion), u_2 (plan approval by the ACC), u_3 (transfer of the

non-approved plan for revision), u_4 (transfer of a task and a work plan to the executor), u_5 (work completion), and u_6 (entering of the changes made in the IT system library). The following stages of the regulations are considered to be model states: x_1 (formation of a task and an implementation plan), x_2 (coordination of a task and a work plan by the ACC), x_3 (coordination of a task and a work plan by the process manager), x_4 (carrying out works by the executor), x_5 (check of the carried out works by the process manager), and x_6 (completed works). Available information about some stages of the regulations is used as outputs. The model allows noncritical deviations in the actions of the process agents from the original (reference) regulations. The initiator's error $f_1 : (x_2 \to x_3)_{x_1,u_1}$ is considered. The fuzzy covering φ is specified by the membership functions $\mu_{B^0_\varphi} = (1\ 0\ 0\ 0\ 0\ 0)$, $\mu_{B^1_\varphi} = (0\ 1\ 1\ 0\ 0\ 0)$, $\mu_{B^2_\varphi} = (0.1\ 0.1\ 0\ 0.8\ 0\ 0)$, $\mu_{B^3_\varphi} = (0.2\ 0.1\ 0\ 0\ 0.7\ 0)$, and $\mu_{B^4_\varphi} = (0\ 0\ 0\ 0\ 0\ 1)$. Recall that the blocks of the covering correspond to the observer's states x^0_0, x^1_0, x^2_0, x^3_0, and x^4_0.

The reference trajectory has the form $x_1 \to x_2 \to x_3 \to x_4 \to x_5 \to x_6$. This trajectory corresponds to the following sequence of transitions between the blocks of the covering $\varphi : B^0_\varphi \to B^1_\varphi \to B^1_\varphi \to B^2_\varphi \to B^3_\varphi \to B^4_\varphi$, initiated by the sequence of inputs u_1, u_2, u_4, u_5, u_6 and leading to the sequence of outputs $y_1, y_2, y_3, y_4, y_2, y_5$ (see [14, Table 4]).

The trajectory $x_1 \to x_2 \to x_3 \to x_2 \to x_3 \to x_4 \to x_5 \to x_6$ is also correct: the process manager transfers the task and work plan for repeated coordination to the ACC. It corresponds to the following sequence of transitions between the blocks of the covering $B^\varphi : B^0_\varphi \to B^1_\varphi \to B^1_\varphi \to B^1_\varphi \to B^1_\varphi \to B^2_\varphi \to B^3_\varphi \to B^4_\varphi$, initiated by the sequence of inputs u_1, u_2, u_4, u_2, u_4, u_5, u_6 and leading to the sequence of outputs y_1, y_2, y_3, y_2, y_3, y_4, y_2, y_5 (see [14, Table 5]).

The trajectory $x_1 \to x_3 \to x_4 \to x_5 \to x_6$ is incorrect, caused by an error of the process initiator. It corresponds to the following sequence of transitions between the blocks of the covering φ: $B^0_\varphi : B^1_\varphi \to B^2_\varphi \to B^3_\varphi \to B^4_\varphi$, initiated by the sequence of inputs u_1, u_2, u_4, u_5 and leading to the sequence of outputs y_1, y_3, y_4, y_2, y_5 (see [14, Table 6]). If this error is detected, it is proposed to make a transition to the reference trajectory, starting from the state $x_2, x_2 \in B^1_\varphi$, using the input u_4.

The real-time formation algorithm for the new input u_* and the mapping g ensuring accommodation to the process initiator's error can be obtained based on the results of this paper. It is omitted here because of simplicity and page limits.

8. CONCLUSIONS

In this paper, the following results have been obtained when solving the general fault accommodation problem for DESs.

1. The mathematical apparatus of the pair algebra of coverings previously proposed by the authors [17] has been extended to the case of covering blocks specified by fuzzy sets. The pair algebra of fuzzy coverings has been elaborated. The necessary computations of the operations and operators of the algebra can be supported by existing applied software packages, e.g., MATLAB.

2. The concept of an observer as applied to an FFSA has been introduced. New mathematical constructs of the pair algebra of fuzzy coverings have been used to develop an algorithm for designing an FFSA observer (Algorithm 4).

3. An optimization approach has been proposed to find the new input u_*, mapping g, and input u of the DES in order to ensure a high degree of implementability of the fault-tolerant control law.

FUNDING

This work was supported by the Russian Science Foundation, project no. 25-19-00590, https://rscf.ru/project/25-19-00590/.

REFERENCES

1. Blanke, M., Kinnaert, M., Lunze, J., and Staroswiecki, M., *Diagnosis and Fault-Tolerant Control*, Berlin: Springer-Verlag, 2003.
2. Edwards, C., Spurgeon, S.K., and Patton, R.J., Sliding Mode Observers for Fault Detection and Isolation, *Automatica*, 2000, vol. 36, no. 4, pp. 541–553.
3. Feng, W., Li, M., Zhang, W., et al., Sliding Mode Fault-Tolerant Control Based on Fast Convergence Law for Faults of PMSM, *Chinese Journal of Electrical Engineering*, 2024, vol. 10, no. 3, pp. 147–157.
4. Zhirabok, A.N. and Shumsky, A.E., Fault Accommodation in Nonlinear Dynamic Systems, *Journal of Computer and Systems Sciences International*, 2009, vol. 48, no. 4, pp. 559–566.
5. Zhirabok, A.N., Kotta, Ü., and Shumsky, A.E., Accommodation to Defects in the Discrete Dynamic Systems, *Automation and Remote Control*, 2014, vol. 75, no. 6, pp. 977–990.
6. Wonham, W.M., Cai, K., and Rudie, K., Supervisory Control of Discrete-Event Systems: A Brief History — 1980–2015, *IFAC-PapersOnLine*, 2017, vol. 50, no. 1, pp. 1791–1797.
7. Cassandras, Ch. and Lafortune, S., *Introduction to Discrete Event Systems*, 2nd ed., New York: Springer Science+Business Media, 2008.
8. Zaytoon, J. and Lafortune, S., Overview of Fault Diagnosis Methods for Discrete Event Systems, *Annual Reviews in Control*, 2013, vol. 37, no. 2, pp. 308–320.
9. Shumskii, A.E. and Zhirabok, A.N., Method of Accommodation to the Defects in Finite Automata, *Automation and Remote Control*, 2010, vol. 71, no. 5, pp. 837–846.
10. Hartmanis, J. and Stearns, R., *Algebraic Structure Theory of Sequential Machines*, Englewood Cliffs, New Jersey: Prentice-Hall, 1966.
11. Zadeh, L.A., Fu, K.S., Tanaka, K., and Shimura, M., *Fuzzy Sets and Their Applications to Cognitive and Decision Processes*, New York: Academic Press, 1975.
12. Wee, W.G. and Fu, K.S., A Formulation of Fuzzy Automata and Its Application as a Model of Learning Systems, *IEEE Trans. Syst. Science and Cybernetics*, 1969, vol. SSC-5, no. 3, pp. 215–223.
13. Ulzutuev, I.E. and Maximov, A.A., The Properties of Subautomata Lattices of Fuzzy Semiautomata and Their Determinizators, *Vestnik of Saratov State Technical University*, 2015, vol. 79, no. 2, pp. 117–126.
14. Shumsky, A.E. and Zhirabok, A.N., A Fault Diagnosis Method for Discrete-Event Systems Based on the Fuzzy Finite State Automaton Model, *Control Sciences*, 2025, no. 2, pp. 31–41.
15. Borisov, V.V., Kruglov, V.V., and Fedulov, A.S., *Nechetkie modeli i seti* (Fuzzy Models and Networks), 2nd ed., Moscow: Goryachaya Liniya – Telekom, 2012.
16. Voevodin, V.V. and Kuznetsov, Yu.A., *Matritsy i vychisleniya* (Matrices and Calculations), Moscow: Nauka, 1984.
17. Zhirabok, A.N., Kalinina, N.A., and Shumskii, A.E., Method for the Functional Diagnosis of Non-Deterministic Finite Automata, *Journal of Computer and Systems Sciences International*, 2020, vol. 59, no. 4, pp. 565–574.

This paper was recommended for publication by N.I. Selvesyuk, a member of the Editorial Board