A FAULT DIAGNOSIS METHOD FOR DISCRETE-EVENT SYSTEMS BASED ON THE FUZZY FINITE STATE AUTOMATON MODEL

A. E. Shumsky* and A. N. Zhirabok**

*,**Far Eastern Federal University, Vladivostok, Russia

*✉ a.e.shumsky@yandex.com, **✉ zhirabok@mail.ru

Abstract. This paper considers the problem of fault diagnosis in critical-purpose discrete-event systems described by the fuzzy finite state automaton (FSA) model. A solution method involving the mathematical apparatus of fuzzy logic is proposed. Fuzzy logic operations are described, and the concept of the determinizer of a fuzzy FSA is introduced. A diagnosis scheme that forms a structured residual vector is given. This scheme contains several channels (according to the number of possible faults in the system). Each channel is based on an observer, i.e., a determinizer of a special fuzzy FSA that simultaneously considers the possibility of both correct and incorrect transitions of the automaton (the normal operation of the system and the occurrence of a system fault, respectively). Another part of the channel is the decision block. Some ways to design the observer and the decision block are proposed. The features of the solution method are illustrated on the example of error monitoring for human operators in IT systems.

Keywords: discrete-event systems, fuzzy logic, fuzzy finite state automata, determinizer, fault diagnosis, IT systems, monitoring.

INTRODUCTION

Strict requirements for the reliability and fault tolerance of modern complex critical-purpose systems require implementing their diagnosis, i.e., the detection and isolation of faults arising during system operation, to parry or eliminate them in due time. This paper is devoted to the problem of fault diagnosis in the so-called discrete-event systems (DESs). Note that DESs include not only systems that naturally belong to this class (e.g., digital information processing and control systems).

Many systems traditionally classified as continuous (such as physical, technical (including man-machine), and socio-economic) can be treated as DESs at the top level of their hierarchy. The distinctive features of DESs are as follows [1]:

- Discrete-event systems have discrete time and discrete values of state variables.
- The state space of a DES is finite; for example, possible states are idleness, operation in a certain mode, malfunctioning, recovery, etc.
- DES operation is determined by events that can be consequences of various commands, e.g., “start operation,” “change operation mode,” “perform DES diagnosis,” “start DES recovery,” “complete DES recovery,” “complete DES operation,” etc.
- As a rule, a discrete-event system behaves randomly due to realizing (possibly) different transitions from one state to another initiated by the same event.

Figure 1 presents a generalized scheme of fault diagnosis and fault-tolerant control of a DES. According to this scheme, a controller generates external events for the system (DES inputs or commands). The DES responds by forming its internal events, considered to be DES outputs. The controller monitors the DES outputs and the diagnosis determined by a diagnosis system (DS) to produce a new external event (DES input). In turn, the DS monitors both external and internal events to determine the diagnosis. The latter answers the question: Is the DES in good condition? If the answer is negative, an additional judgment will be made on the type of fault. If the fault is detected and classified, the controller will report a new event, and the response will be to parry this fault (form a command sequence mitigating the fault’s effect on the achievement of the system goal) or eliminate this fault (repair the DES).

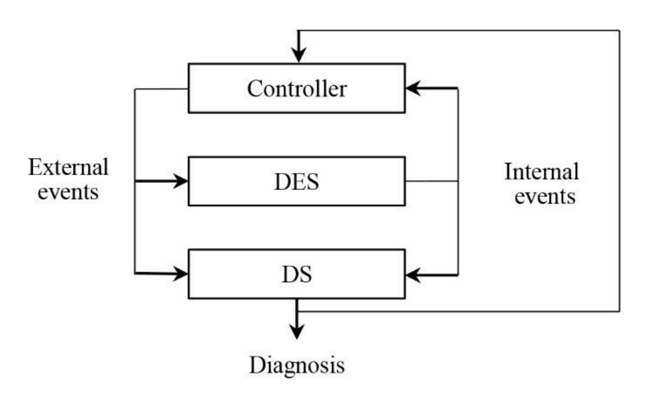

Fig. 1. The scheme of fault diagnosis and fault-tolerant control of DESs.

A fairly complete review of the existing DES diagnosis methods can be found in the papers [2, 3] and monograph [4]. As noted in [3], the following mathematical models are most widespread to describe DESs:

– deterministic finite state automata (FSA),
– probabilistic FSA and Markov chains,
– Petri nets.

This paper focuses exclusively on the application of FSA models. With such models used to diagnose DESs, 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.

Whenever no deterministic FSA model of a DES is available, in addition to probabilistic FSA, nondeterministic and fuzzy FSA can be used. As compared to probabilistic FSA models, nondeterministic and fuzzy models allow considerably reducing the volume of necessary calculations, thereby accelerating the fault diagnosis process. Nondeterministic and fuzzy FSA models cover the situation when transitions to different states can be realized for a fixed state and fixed input of an automaton. For nondeterministic FSA, it is impossible to give priority to the realization of a certain transition; for fuzzy FSA, however, additional (e.g., statistical) information, expert assessments, training results, etc. can be utilized to talk about the degree of confidence in the realization of each possible transition.

For diagnosing DESs described by the nondeterministic FSA model, methods based on pairwise partition algebra and pairwise covering algebra were presented in [5] and [6], respectively; also, see [7].

The objective of this paper is to develop a new fault diagnosis method based on the fuzzy FSA model. We involve mathematical constructs of fuzzy logic [8] as well as the concepts of a fuzzy finite state automaton [9] and its determinizer [10].

The features of this method are illustrated on the example of error monitoring for human operators in IT systems; it was previously considered in [5] and [6] for fault diagnosis within the nondeterministic FSA model.

1. THE OBJECTIVES AND STRUCTURE OF THIS PAPER

1.1. Models Used

Let a DES in good condition be described by the fuzzy FSA model

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

where $U=\{u_1, u_2, \ldots, u_m\}$, $X=\{x_1, x_2, \ldots, x_n\}$, and $Y=\{y_1, y_2, \ldots, y_l\}$ denote the finite sets of inputs, states, and outputs, respectively; $x(0)\in X$ is a known initial state; $\delta: X\times U\to \mu(X)$ is a fuzzy transition function; $\mu(X)\in\{\mu(x_i)\in[0,1],\ 1\le i\le n\}$ is a fuzzy set; finally, $\lambda: X\to Y$ is an output function.

We utilize the matrix representation for the fuzzy transition function δ: the transitions performed under an input u_k are described by a matrix M^k of dimensions $n\times n$, in which each element $M^k_{i,j}\in[0,1]$ characterizes the degree of confidence that, given $u_k\in U$, an automaton A will move from a state $x_i\in X$ to a state $x_j\in X$. Let $S(U)$ denote the set of all matrices $M^k, 1\le k\le m$.

If it is impossible to specify the degree of confidence in transitions reasonably, we propose to proceed as follows: set an element of the matrix $M^k_{i,j}$ corresponding to an admissible transition equal to 1 whereas the element of this matrix corresponding to an inadmissible transition equal to 0. Thereby, one passes from the fuzzy FSA model of a DES to its nondeterministic counterpart.

We specify the output function λ using a matrix L of dimensions $l\times n$ in which $L_{i,j}=1$ if the output $y_i\in Y$ is generated by the automaton A in the state $x_j\in X$ and $L_{i,j}=0$ otherwise. Let $S(Y)$ denote the set of all rows $\{L_i,\ 1\le i\le l\}$ of the matrix L.

Consider $X_{i,k}$, $X_{i,k}\subseteq X$, the set of all states reachable from a state $x_i\in X$ under an input $u_k\in U$. Assume that a fault $f_s, 1\le s\le N$, in the DES model (1) can be represented by a distortion of the transition function δ such that, given $u_k \in U$, an inadmissible transition from a state x_i to a state $x_t \notin X_{i,k}$ is realized instead of an admissible transition from the former state to a state $x_j \in X_{i,k}$. This fact will be indicated by $f_s : \left(x_j \to x_t\right)_{i,\,k}$. In this case, the matrix M^k of the faulty DES will be obtained by changing the value of the element $M_{i,t}^k$ of the matrix M^k of the operable DES from 0 to 1. To simplify the presentation, we accept the hypothesis of single faults from a predetermined list $F = \{f_1, f_2, \ldots, f_N\}$. Let A_s denote an auxiliary automaton whose transition function δ_s is obtained by changing (in the above manner) the transition function δ of the automaton A to the transition caused by the fault f_s:

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

Due to the construction procedure of the matrix M^k, model (2) covers the possibility of a "correct" transition in the operable DES and, moreover, the possibility of an "incorrect" transition to the state caused by the DES fault f_s. Therefore, there is a definite correspondence between the behavior of model (2) and the behavior of both the operable and faulty DESs. In subsection 3.2, we propose a method for evaluating this correspondence in terms of possibility (confidence) and form a structured residual vector based on the method.

1.2. Fault Diagnosis Scheme

To detect and isolate faults, the idea is to use the fault diagnosis scheme shown in Fig. 2. This scheme contains N channels (according to the number of possible DES faults), and each channel includes a deterministic FSA A_s^d and a decision block DB_s. For the diagnosis scheme design, it is necessary to determine the components of each channel. Therefore, the remainder of this paper is organized as follows.

1.3. The Structure of This Paper

Section 2 provides the mathematical constructs of fuzzy logic required to obtain the main results of the paper. In Section 3, the design problem of a deterministic FSA A_s^d (called the observer of the nondeterministic FSA A_s) is reduced to the modified problem of finding the determinizer [10] of this automaton. Also, we propose an operation rule for DB_s, $1 \le s \le N$, ensuring the structuredness of the residual vector, an important property with the following essence. First, the zero value of all the components of the residual vector means the absence of DES faults. Second, if only one component of the residual vector is 0 (the others being equal to 1), then the DES has a fault with the number coinciding with that of the zero component of the residual vector. In Section 4, we consider a numerical example and simulation results to illustrate the features of the method proposed. The outcomes of this paper are summarized in the Conclusions.

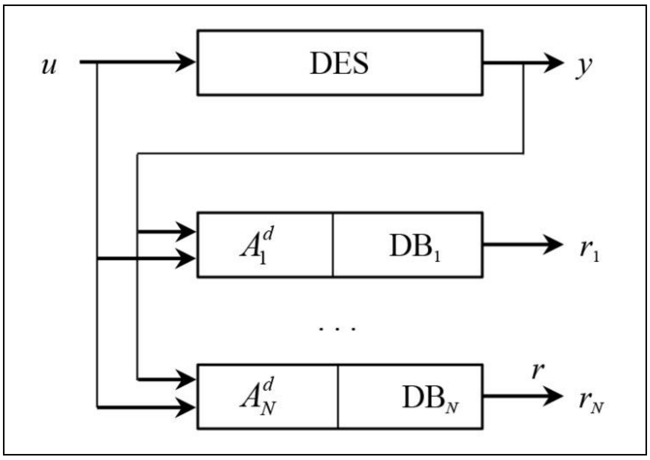

Fig. 2. The fault diagnosis scheme of DESs.

2. MATHEMATICAL CONSTRUCTS

2.1. Fuzzy Logic Operations

Recall several operations of fuzzy logic [8], playing an important role for the further presentation. A fuzzy matrix $B = \{B_{ij}\}$ is a matrix with elements $B_{ij} \in [0,1]$. Hence, the above matrices M^k, $1 \le k \le m$, are fuzzy. 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 by [10]

$$\left(BC\right)_{ij} = \max \min_{i,\,j}\left(B_{ih}\ C_{hj}\right),$$

where B_{ih} and C_{hj} are the corresponding elements of matrices B and C, $1 \le i \le a$, $1 \le h \le b$, and $1 \le j \le c$. This formula generalizes the well-known matrix multiplication [11, *p. 24*]; it is obtained by replacing the product of matrix elements by the operation of finding the minimum and the sum of elements by the operation of finding the maximum. Here is a simple numerical example.

Example 1. Consider matrices B and C of the form

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

The calculations yield the following results: $(BC)_{11} = \max\left(\min(0.1\ \ 0.6),\ \min(0.9\ \ 0.4)\right) = 0.4$, $(BC)_{12} = \max\left(\min(0.1\ \ 0.3),\ \min(0.9\ \ 0.8)\right) = 0.8$, $(BC)_{21} = \max\left(\min(0.7\ \ 0.6),\ \min(0.2\ \ 0.4)\right) = 0.6$, $(BC)_{22} = \max\left(\min(0.7\ \ 0.3),\ \min(0.2\ \ 0.8)\right) = 0.3$. Thus, the product of the fuzzy matrices B and C is

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

2.2. The Concept of the Determinizer of a Fuzzy FSA

The definition of the determinizer of a fuzzy FSA [10] is introduced as follows. Let E be some set of fuzzy n-dimensional row vectors whose components can take values on the interval $[0, 1]$. Also, let E^0, $E^0 \subseteq E$, be the set of all unit n-dimensional row vectors, called the generating states of the determinizer $D(A')$. The closure $[E]_\Sigma$ of the set E with respect to a signature (set of admissible operations) Σ is the set of all vectors, including those from E, that can be obtained by applying operations from the signature Σ to vectors from E. We construct the closure using the following procedure.

Step 1. Assign $i = 0$.

Step 2. Find the vector set $E^{i+1} = E^i \cup \left[E^i\right]_\Sigma$.

Step 3. If $E^{i+1} = E^i$, assign $[E]_\Sigma = E^i$ and complete the procedure.

Step 4. Otherwise, let $i = i+1$ and go back to Step 2.

Let $A' = (U, X, \delta)$ be a fuzzy semiautomaton (i.e., an automaton A without output function) and $S(U) = \left\{M^k, 1 \le k \le m\right\}$ be the set of fuzzy transition matrices of the automaton A. Assume that the signature Σ includes all operations involving the multiplication of an n-dimensional row vector on the right by a matrix from the set $S(U)$. The determinizer of a fuzzy semiautomaton A' is a deterministic FSA described by the triple

$$D(A') = \left(S(X), U, \Delta\right),$$

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

$$\Delta(\mu_i, u_k) = \mu_i \cdot M^k,\ \mu_i \in S(X),\ M^k \in S(U). \quad (3)$$

To explain formula (3), we recall that the matrix M^k describes a transition activated by an input u_k.

Example 2. Consider a semiautomaton A' defined by the fuzzy transition matrices

$$M^1 = \begin{pmatrix} 0.4 & 0.6 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix},\ M^2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Let the generating states of the determinizer be $\mu_1 = (1\ \ 0\ \ 0)$, $\mu_2 = (0\ \ 1\ \ 0)$, and $\mu_3 = (0\ \ 0\ \ 1)$. The necessary calculations by formula (3) yield the following vectors μ_4, μ_5, and μ_6 of the closure $[E]_{S(U)}$: $\mu_4 = \mu_1 \cdot M^1 = (0.4\ \ 0.6\ \ 0)$, $\mu_5 = \mu_4 \cdot M^1 = (0.4\ \ 0.4\ \ 0.6)$, $\mu_6 = \mu_4 \cdot M^2 = (0.6\ \ 0\ \ 0.4)$. Following similar considerations, we obtain $\mu_7 = (0.4\ \ 0.6\ \ 0.4)$, $\mu_8 = (0.6\ \ 0.4\ \ 0.4)$, $\mu_9 = (0\ \ 0.4\ \ 0.6)$, $\mu_{10} = (0\ \ 0\ \ 0.4)$, $\mu_{11} = (0\ \ 0.4\ \ 0)$, $\mu_{12} = (0.4\ \ 0\ \ 0)$, $\mu_{13} = (0.4\ \ 0.4\ \ 0)$, $\mu_{14} = (0.4\ \ 0.4\ \ 0.4)$, $\mu_{15} = (0.4\ \ 0\ \ 0.4)$, $\mu_{16} = (0\ \ 0.4\ \ 0.4)$.

Finally, taking $M(X) = \left\{\mu_i, 1 \le i \le 16\right\}$, we construct the transition table of the determinizer $D(A')$ (Table 1). For example, it indicates the transitions from the state μ_9 to the state μ_{10} under the input u_1 and those from the state μ_9 to the state μ_4 under the input u_2 since $\mu_9 \cdot M^1 = \mu_{10}$ and $\mu_9 \cdot M^2 = \mu_4$. Note that the number of states of the determinizer $D(A')$ significantly exceeds that of the original semiautomaton A'. Indeed, when constructing the determinizer, we preserve all information about the degree of confidence in realizing each possible transition since each determinizer's state is a vector of possible DES transitions to the corresponding states at a definite time instant. ♦

Table 1

The transition table of the determinizer $D(A')$

	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}	μ_{11}	μ_{12}	μ_{13}	μ_{14}	μ_{15}	μ_{16}
u_1	μ_4	μ_3	μ_3	μ_5	μ_5	μ_7	μ_5	μ_7	μ_{10}	μ_{10}	μ_{10}	μ_{13}	μ_{14}	μ_{14}	μ_{14}	μ_{10}
u_2	μ_3	μ_1	μ_2	μ_6	μ_7	μ_9	μ_8	μ_5	μ_4	μ_{11}	μ_{12}	μ_{10}	μ_{15}	μ_{14}	μ_{16}	μ_{13}

To obtain an upper bound for the number of determinizer's states, we consider that the components of all vectors from the set $S(X)$ can take only the values contained in the matrices from the set $S(U)$. Excluding the zero vector from the analysis gives the formula

$$\#S(X) \le q^n - 1, \tag{4}$$

where # denotes the set cardinality; q is the number of different values for the elements of fuzzy matrices from the set $S(U)$; as before, n stands for the number of states of the fuzzy FSA. For example, consider system (3); in this case, we have $q = 4$ (the set of different values for the elements of the matrices M^1 and M^2 includes the numbers $\{0;\ 0.4;\ 0.6;\ 1\}$), $n = 3$, and, consequently, $\#S(X) \le 4^3 - 1 = 63$.

Example 3. The above approach can be applied to the determinization of nondeterministic FSA. As an illustration, we consider the nondeterministic FSA model obtained from the fuzzy one (3) by setting the unit degrees of confidence for all admissible transitions:

$$M^1 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, M^2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \tag{5}$$

In the case of nondeterministic FSA, the dimension of the resulting determinizer is not necessarily smaller than that of the original automaton. Indeed, for nondeterministic FSA we have $q = 2$ and, by formula (4), the number of determinizer's states is

$$\#S(X) \le 2^n - 1.$$

Particularly for the nondeterministic FSA (5), it follows that $\#S(X) \le 2^3 - 1 = 7$. Omitting intermediate calculations, we directly present the transition table of the determinizer of nondeterministic FSA (5).

In Table 2, the determinizer's states are $\mu_1 = (1 \quad 0 \quad 0)$, $\mu_2 = (0 \quad 1 \quad 0)$, $\mu_3 = (0 \quad 0 \quad 1)$, $\mu_4 = (1 \quad 1 \quad 0)$, $\mu_5 = (1 \quad 1 \quad 1)$, $\mu_6 = (1 \quad 0 \quad 1)$, and $\mu_7 = (0 \quad 1 \quad 1)$. ♦

Table 2

The transition table of the determinizer of the nondeterministic FSA (5)

	μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7
u_1	μ_4	μ_3	μ_3	μ_5	μ_5	μ_5	μ_3
u_2	μ_3	μ_1	μ_2	μ_6	μ_5	μ_7	μ_4

Note that the algebraic determinization approach for nondeterministic FSA based on partition algebra [5] and covering algebra [6] surely yields a deterministic FSA with a dimension not exceeding that of the original automaton. The reason is that the algebraic approach does not require preserving information about the degree of confidence in the realization of each possible transition. The fuzzy approach under consideration involves additional useful information about the degree of confidence in the realization of each possible transition during the fault diagnosis procedure, providing a potential opportunity to increase (if necessary) the depth of fault isolation. The price for this is a significant dimension of the determinizer's transition table.

The main difference between the descriptions of the deterministic FSA A_s^d, $1 \le s \le N$, (the fault diagnosis scheme in Fig. 2) and the determinizer $D(A_s')$ of the corresponding semiautomaton A_s' is that the transition function of the automaton A_s^d additionally depends on the outputs of the original automaton. As a result, additional information can be used to adjust the behavior and reduce the dimension of the automaton A_s^d. Drawing an analogy with the observers of a continuous dynamic system, we call the deterministic FSA A_s^d an observer of the fuzzy FSA A_s.

3. DIAGNOSIS CHANNEL DESIGN

3.1. The Observer A_s^d

The further presentation concerns the channel of the fault diagnosis scheme intended to isolate a fault f_s. First, consider a method for finding the transition function δ_s^d of the observer A_s^d based on a slight modification of the relation (3). For vectors $\mu_i \in S(X)$ and $L_j \in S(Y)$, we introduce the notation

$$\langle \mu_i,\ L_j \rangle = \big(\min(\mu_{i,1},\ L_{j,1}),\ \min(\mu_{i,2},\ L_{j,2}), \ldots, \min(\mu_{i,n},\ L_{j,n})\big). \tag{6}$$

Let the transition function $\Delta : S(X) \times S(Y) \times S(U) \to S(X)$ of the deterministic FSA A_s^d be defined as follows:

$$\Delta(\mu_i,\ y_j,\ u_k) = \langle \mu_i,\ L_j \rangle \cdot M^k, \quad \mu_i \in S(X),\ L_j \in S(Y),\ M^k \in S(U). \tag{7}$$

With $\langle \mu_i, L_j \rangle$ used in formula (7) instead of μ_i in the determinizer's transition function (3), we specify the values of membership functions (to the corresponding fuzzy sets) for the components of the vector $x \in X$ before the transition.

Example 4. Consider a fuzzy FSA described by the matrices M^1 and M^2 of Example 2 (in the absence of faults). Let the output function of this automaton be given by the matrix

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}.$$

Its transition errors are a consequence of two faults, $f_1 : (x_3 \to x_1)_{1,2}$ and $f_2 : (x_3 \to x_2)_{1,2}$. The automaton's matrices M_1^2 and M_2^2 including an additional erroneous transition due to an appropriate fault (indicated by the matrix subscript) have the form

$$M_1^2 = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}, \; M_2^2 = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

The matrix M^1 is unaffected by the fault and retains its original form.

Now we design the observer A_1^d (in this case, using only the matrix M_1^2). As in the previous example, let the generating states be $\mu_1 = (1 \;\; 0 \;\; 0)$, $\mu_2 = (0 \;\; 1 \;\; 0)$, and $\mu_3 = (0 \;\; 0 \;\; 1)$. Calculations according to the right-hand side of the relation (7) give

$$\langle \mu_1, L_1 \rangle \cdot M^1 = (\min(1,1) \quad \min(0,0) \quad \min(0,0))$$

$$\times \begin{pmatrix} 0.4 & 0.6 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} = (0.4 \quad 0.6 \quad 0) = \mu_4.$$

By analogy, we find $\mu_5 = (1 \;\; 0 \;\; 1)$, $\mu_6 = (0.4 \;\; 0.4 \;\; 0)$, $\mu_7 = (0 \;\; 0 \;\; 0.6)$, $\mu_8 = (0.4 \;\; 0 \;\; 0.4)$, $\mu_9 = (0.6 \;\; 0 \;\; 0)$, $\mu_{10} = (0 \;\; 0 \;\; 0.4)$, $\mu_{11} = (0.4 \;\; 0 \;\; 0)$, $\mu_{12} = (0 \;\; 0.6 \;\; 0)$, $\mu_{13} = (0 \;\; 0.4 \;\; 0)$, and $\mu_{14} = (0.6 \;\; 0 \;\; 0.6)$ and build the transition table of the observer A_1^d (Table 3).

In this table, dashes indicate the transitions corresponding to the incompatible values of the observer state and DES output. For example, the observer state μ_2 allows only the DES output y_2, which is immediate from the expressions for the vector μ_2 and the matrix L. Hence, the combination of μ_2 and y_1 is impossible during the faultless operation of the DES.

Finally, we compare the dimension of the determinizer $D(A')$ (Table 1) and the dimension of the observer A_1^d (Table 3), emphasizing that the latter does not exceed the former. However, in practice, the observer may have a higher dimension than the original fuzzy FSA. ♦

Seemingly, this fact should limit the practical realizability of the fault diagnosis procedure of DESs based on the fuzzy FSA model due to the significant dimension of real systems. As expected, these limitations should be less pronounced within the algebraic approach [5, 6]. Meanwhile, the actual things differ. The point is that the algebraic approach finds a tabular description for the transition function of observers (deterministic FSA A_i^d, $1 \le i \le N$) from the tabular description of the original finite state automaton model of the DES. In contrast, the approach proposed specifies the transition function in a compact analytical form, being therefore insensitive to the growth of the dimension of the original DES model. Moreover, with this feature, the approach proposed is preferable when designing fault diagnosis schemes for DESs based on the deterministic and nondeterministic FSA models. It successfully overcomes the so-called "curse of dimensionality," inevitably arising for all methods with a tabular (or graph-based) description of the FSA model of DESs.

3.2. The Decision Block of the Diagnosis Channel

Assume that during the fault diagnosis procedure, the DES output y_j and the state μ_i of the observer A_s^d are generated simultaneously. Let σ_s denote the value of the maximum component of the vector $\langle \mu_i, L_j \rangle$. This value will be regarded as the degree of

Table 3

The transition table of the observer A_1^d

		μ_1	μ_2	μ_3	μ_4	μ_5	μ_6	μ_7	μ_8	μ_9	μ_{10}	μ_{11}	μ_{12}	μ_{13}	μ_{14}
u_1	y_1	μ_4	-	-	μ_6	μ_4	μ_6	-	μ_6	μ_4	-	μ_6	-	-	μ_4
	y_2	-	μ_3	μ_3	μ_7	μ_3	μ_{10}	μ_7	μ_{10}	-	μ_{10}	-	μ_7	μ_{10}	μ_7
u_2	y_1	μ_5	-	-	μ_8	μ_5	μ_8	-	μ_8	μ_{14}	-	μ_8	-	-	μ_{14}
	y_2	-	μ_1	μ_2	μ_{12}	μ_2	μ_{11}	μ_{12}	μ_{13}	-	μ_{13}	-	μ_{11}	μ_{13}	μ_{12}

confidence that the DES behavior corresponds to the behavior of the observer A_s^d. In view of the aforesaid, we specify the relation Ψ_s:

$$\left(\mu_i,\ y_j\right) \in \Psi_s \Leftrightarrow \sigma_s \neq 0. \qquad (8)$$

Assume now that the relation Ψ_s (8) has been verified as false for a particular pair y_j and μ_i (i.e., $\sigma_s = 0$). This indicates the presence of an error (fault) $f_k,\, k \neq s$, in the DES. In this case, the corresponding residual value is $r_s = 1$; otherwise ($\sigma_s \neq 0$), $r_s = 0$. To judge unambiguously the fault type in the DES based on the structured residual vector, only one of its components should retain the zero value. (The number of this component will be the fault number.) If not (several residuals take zero values), then extra checks are required, e.g., using additional measurements and/or special tests [12]. Such checks should be carried out sequentially for all faults in the order of decreasing degrees of confidence from the list $\{\sigma_1, \sigma_2, ..., \sigma_N\}$.

4. AN ILLUSTRATIVE EXAMPLE

The change management process in an IT system was described in detail in [5, *Fig. 1*; 6], including a method for obtaining its deterministic and nondeterministic FSA models. Note that this process is one of the most important for IT systems: it is responsible for managing the lifecycle of all changes and facilitates the implementation of useful changes with minimum interruption of IT services. The change management process involves the following participants:

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

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

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

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

The transition and output tables of the deterministic and nondeterministic FSA models of the process were also presented in the papers cited. The deterministic FSA model describes the actions of the process participants in full compliance with the regulations prescribed. In practice, a nondeterministic FSA should be used as the initial model due to some (non-critical) deviations of the participants' actions from the prescribed regulations, allowed to clarify the regulations or the participants' knowledge of them. In particular, the following situations were considered:

– The process manager sends the reviewed result of a completed task to the ACC for re-approval.

– The process manager sends a received and agreed task back to the initiator.

– The executor sends a received and agreed task back to the initiator.

– The executor sends a received and agreed task back to the ACC for approval.

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 (see the output function below).

The fuzzy FSA model of the change management process in an IT system that describes the error-free work of all process participants (the initiator, executor, and manager) can be obtained based on the nondeterministic counterpart [6, *Table 3*] and the available statistical information about the possible actions of the participants when following the regulations:

$$M^1 = \begin{pmatrix} 0 & 1 & 0 & & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$M^2 = \begin{pmatrix} 0 & 0 & & 0 & 0 & & 0 \\ 0 & 0 & & 1 & 0 & \cdots & 0 \\ 0 & 0 & \vdots & 0 & 0 & & 0 \\ 0 & 0 & & 0 & 0 & \ddots & \vdots \\ 0 & 0 & & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$M^3 = \begin{pmatrix} 0 & 0 & 0 & & 0 \\ 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & & 0 \\ & \vdots & & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$M^4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1 & 0.1 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$M^5=\begin{pmatrix}0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0.2&0.1&0&0&0.7&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{pmatrix},$$

$$M^6=\begin{pmatrix}0&0&\cdots&0&0\\&\vdots&\ddots&&\vdots\\0&0&\cdots&0&1\\0&0&&0&0\end{pmatrix}.$$

All these matrices have dimensions 6×6. The output function is described by the matrix

$$L=\begin{pmatrix}1&0&0&0&0&0\\0&1&0&0&1&0\\0&0&1&0&0&0\\0&0&0&1&0&0\\0&0&0&0&0&1\end{pmatrix}.$$

As in [6], we consider the errors of the initiator, executor, and process manager: $f_1:(x_2\to x_3)_{(x_1,u_1)}$, $f_2:(x_5\to x_6)_{(x_4,u_5)}$, and $f_3:(x_3\to x_4)_{(x_2,u_2)}$, respectively. The corresponding matrices are

$$M_1^1=\begin{pmatrix}0&1&1&&0\\0&0&0&\cdots&0\\0&0&0&&0\\&\vdots&&\ddots&\vdots\\0&0&0&\cdots&0\end{pmatrix},$$

$$M_2^5=\begin{pmatrix}0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0.2&0.1&0&0&0.7&1\\0&0&0&0&0&0\\0&0&0&0&0&0\end{pmatrix},$$

$$M_3^2=\begin{pmatrix}0&0&0&0&0&0\\0&0&1&1&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{pmatrix}.$$

All other matrices retain their values.

The tables below present the simulations for different admissible results of the participants to the change management process as well as the errors of the initiator, executor, and process manager. The residuals and auxiliary variables were generated using the relations (6)-(8) and the above matrices. The matrices used are: M_1^1, M^2, M^3, M^4, M^5, and M^6 (the first channel); M^1, M^2, M^3, M^4, M_2^5, and M^6 (the second channel); M^1, M_3^2, M^4, M^5, and M^6 (the third channel). The following scenarios were simulated.

Scenario 1. The process regulations are implemented without any deviations (Table 4).

Scenario 2. The process regulations are implemented with an admissible deviation due to the manager's transfer of task formation and work plan for re-approval by the ACC (Table 5).

Scenario 3. The process regulations are implemented with errors made by the initiator, executor, and process manager, respectively (Table 6).

Table 4

Scenario 1

Characteristics		Initial value	Input				
			u_1	u_2	u_4	u_5	u_6
System	State x	x_1	x_2	x_3	x_4	x_5	x_6
	Output y	y_1	y_2	y_3	y_4	y_2	y_5
Channel 1	State μ	μ_{10}	μ_{11}	μ_{12}	μ_{14}	μ_{15}	μ_{16}
	Confidence σ_1	1	1	1	0.8	0.7	0.7
	Residual r_1	0	0	0	0	0	0
Channel 2	State μ	μ_{20}	μ_{21}	μ_{22}	μ_{24}	μ_{25}	μ_{26}
	Confidence σ_2	1	1	1	0.8	0.7	0.7
	Residual r_2	0	0	0	0	0	0
Channel 3	State μ	μ_{30}	μ_{31}	μ_{32}	μ_{34}	μ_{35}	μ_{36}
	Confidence σ_3	1	1	1	0.8	0.7	0.7
	Residual r_3	0	0	0	0	0	0

Table 5

Scenario 2

Characteristics		Initial value	Input						
			u_1	u_2	u_4	u_2	u_4	u_5	u_6
System	State x	x_1	x_2	x_3	x_2	x_3	x_4	x_5	x_6
	Output y	y_1	y_2	y_3	y_2	y_3	y_4	y_2	y_5
Channel 1	State μ	μ_{10}	μ_{11}	μ_{12}	μ_{14}	μ_{12}^*	μ_{14}^*	μ_{15}^*	μ_{16}^*
	Confidence σ_1	1	1	1	0.1	0.1	0.1	0.1	0.1
	Residual r_1	0	0	0	0	0	0	0	0
Channel 2	State μ	μ_{20}	μ_{21}	μ_{22}	μ_{24}	μ_{22}^*	μ_{24}^*	μ_{25}^*	μ_{26}^*
	Confidence σ_2	1	1	1	0.1	0.1	0.1	0.1	0.1
	Residual r_2	0	0	0	0	0	0	0	0
Channel 3	State μ	μ_{30}	μ_{31}	μ_{32}	μ_{34}	μ_{32}^*	μ_{34}^*	μ_{35}^*	μ_{36}^*
	Confidence σ_3	1	1	1	0.1	0.1	0.1	0.1	0.1
	Residual r_3	0	0	0	0	0	0	0	0

Table 6

Scenario 3

Characteristics		Initial value	f_1	f_2				f_3	
			Input						
			u_1	u_1	u_2	u_4	u_5	u_1	u_2
System	State x	x_1	x_3	x_2	x_3	x_4	x_6	x_2	x_4
	Output y	y_1	y_3	y_2	y_3	y_4	y_5	y_2	y_4
Channel 1	State μ	μ_{10}	μ_{11}	μ_{11}	μ_{12}	μ_{14}	μ_{15}	μ_{11}	μ_{12}
	Confidence σ_1	1	1	1	1	0.8	0	1	0
	Residual r_1	0	0	0	0	0	1	0	1
Channel 2	State μ	μ_{20}	μ_{21}	μ_{21}	μ_{22}	μ_{24}	μ_{25}	μ_{21}	μ_{22}
	Confidence σ_2	1	0	1	1	0.8	0.8	1	0
	Residual r_2	0	1	0	0	0	0	0	1
Channel 3	State μ	μ_{30}	μ_{31}	μ_{31}	μ_{32}	μ_{34}	μ_{35}	μ_{31}	μ_{32}
	Confidence σ_3	1	0	1	1	0.8	0	1	1
	Residual r_3	0	1	0	0	0	1	0	0

Note that in different scenarios, each stage of the regulations is implemented a different number of times; this fact is reflected in the tables. The states of the observers of fault diagnosis channels appearing in the tables have the following values: $\mu_{10} = \mu_{20} = \mu_{30} = (1\ 0\ 0\ 0\ 0\ 0)$, $\mu_{11} = (0\ 1\ 1\ 0\ 0\ 0)$, $\mu_{21} = \mu_{31} = (0\ 1\ 0\ 0\ 0\ 0)$, $\mu_{12} = \mu_{22} = (0\ 0\ 1\ 0\ 0\ 0)$, $\mu_{32} = (0\ 0\ 1\ 1\ 0\ 0)$, $\mu_{14} = \mu_{24} = \mu_{34} = (0.1\ 0.1\ 0\ 0.8\ 0\ 0)$, $\mu_{15} = \mu_{35} = (0.2\ 0.1\ 0\ 0\ 0.7\ 0)$, $\mu_{25} = (0.2\ 0.1\ 0\ 0\ 0.7\ 0.8)$, $\mu_{16} = \mu_{26} = \mu_{36} = (0\ 0\ 0\ 0\ 0\ 0.7)$. The states of the observers of fault diagnosis channels appearing in Table 5 have the following values: $\mu_{12}^{*} = \mu_{22}^{*} = (0\ 0\ 0.1\ 0\ 0\ 0)$, $\mu_{32}^{*} = (0\ 0\ 0.1\ 0.1\ 0\ 0)$, $\mu_{14}^{*} = \mu_{24}^{*} = \mu_{34}^{*} = (0.1\ 0.1\ 0\ 0.1\ 0\ 0)$, $\mu_{15}^{*} = \mu_{35}^{*} = (0.1\ 0.1\ 0\ 0\ 0.1\ 0)$, $\mu_{25}^{*} = (0.1\ 0.1\ 0\ 0\ 0.1\ 0.1)$, $\mu_{16}^{*} = \mu_{26}^{*} = \mu_{36}^{*} = (0\ 0\ 0\ 0\ 0\ 0.1)$. The subscripts of the states are interpreted as follows: the first number corresponds to the channel number whereas the second to the input number affecting the transition.

For the sake of convenience, the superscript * indicates the newly appearing states in scenario 2 that are generated by the same inputs as in scenarios 1 and 3. This is due to implementing more steps of the regulations in scenario 2.

According to Tables 4 and 5, both under the full compliance with the regulations and an admissible deviation from the regulations caused by the process manager’s action, zero values of the residuals are formed at the channel outputs. At the same time (see Table 6), when errors occur in the actions of the initiator, executor, and process manager, the resulting structured residual vector allows unambiguously concluding on the error type at the time of its occurrence.

CONCLUSIONS

This paper has proposed a fault diagnosis method for DESs described by the fuzzy FSA model. In comparison with the nondeterministic counterpart, this model can contribute to achieving the required depth of fault diagnosis. Indeed, let the generated residual vector yield no unambiguous conclusion on the fault type. (In other words, the vector contains several zero components.) In this case, several additional checks may be required to localize the fault. To reduce the number of such checks, they should be performed in the order of decreasing the degree of confidence σ_s, $1 \le s \le N$.

Obviously, the method can be extended to simpler models in the form of deterministic and nondeterministic finite state automata. Distinctive features of the method are as follows: no preliminary tabular description of diagnosis means is required; all calculations are carried out directly during the fault diagnosis process using compact analytical relations. This allows overcoming the “curse of dimensionality,” which inevitably arises for the methods with the tabular (or graph-based) description of the FSA model of DESs. Thus, the earlier existing limit on the admissible dimension of the model of the DES diagnosed is almost eliminated. The method proposed can be further developed for the diagnosability analysis and verification of DES models with a large number of states [13].

Acknowledgments. *This work was supported by the Ministry of Science and Higher Education of the Russian Federation, project no. FZNS-2023-0011.*

REFERENCES

1. Wonham, W.M., Cai, K., and Rudle, K., Supervisory Control of Discrete-Event Systems: A Brief History 1980–2015, *Proceedings of the 20th IFAC Congress*, Toulouse, France, 2017, pp. 1827–1833.
2. Sampath, M., Sengupta, R., Lafortune, S., et al., Failure Diagnosis Using Discrete-Event Models, *IEEE Transactions on Control Systems Technology*, 1996, vol. 4, no. 2, pp. 105–124.
3. 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.
4. Cassandras, Ch. and Lafortune, S., *Introduction to Discrete Event Systems*, 2nd ed., New York: Springer Science+Business Media, 2008.
5. Zhirabok, A.N., Kalinina, N.A., and Shumskii, A.E., Technique of Monitoring a Human Operator’s Behavior in Man-Machine Systems, *Journal of Computer and Systems Sciences International*, 2018, vol. 57, no. 3, pp. 443–452.
6. Zhirabok, A.N., Kalinina, N.A., and Shumskii, A.E., Method for the Functional Diagnosis of Nondeterministic Finite State Machines, *Journal of Computer and Systems Sciences International*, 2020, vol. 59, no. 4, pp. 565–574.
7. Hartmanis, J. and Stearns, R., *The Algebraic Structure Theory of Sequential Machines*, New York: Prentice-Hall, 1966.
8. 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.
9. Wee, W.G. and Fu, K.S., A Formulation of Fuzzy Automata and Its Applications as a Model of Learning Systems, *IEEE Trans. Syst. Science and Cybernetics*, 1969, vol. 5, no. 3, pp. 215–223.
10. Ulzutuev, I.E. and Maximov, A.A., The Properties of Subautomata Lattices of Fuzzy Semi-automata and Their Determinizators, *Vestnik of Saratov State Technical University*, 2015, vol. 79, no. 2, pp. 117–126. (In Russian.)
11. Voevodin, V.V. and Kuznetsov, Yu.A., *Matritsy i vychisleniya* (Matrices and Computations), Moscow: Nauka, 1984. (In Russian.)
12. Gruzlikov, A.M. and Kolesov, N.V., Discrete-Event Diagnostic Model for a Distributed Computational System. Independent Chains, *Autom. Remote Control*, 2016, vol. 77, no. 10, pp. 1805–1817.
13. Tuxi, T.M., Carvalho, L.K., Nunes, E.V.L., and Cunha, A.E., Diagnosability Verification Using LTL Model Checking, *Discrete Event Dynamic Systems*, 2022, vol. 32, no. 3, pp. 399–433.

This paper was recommended for publication by S.A. Krasnova, a member of the Editorial Board.

Received March 4, 2025, and revised April 17, 2025. Accepted April 29, 2025.

Author information

Shumsky, Aleksei Evgen'evich. Dr. Sci. (Eng.), Far Eastern Federal University, Vladivostok, Russia
✉ zhirabok@mail.ru
ORCID iD: https://orcid.org/0000-0002-5429-7482

Zhirabok, Aleksei Nilovich. Dr. Sci. (Eng.), Far Eastern Federal University, Vladivostok, Russia
✉ zhirabok@mail.ru
ORCID iD: https://orcid.org/0000-0001-5927-7117

Cite this paper

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* **2**, 31–41 (2025).

Translated into English by *Alexander Yu. Mazurov*,
Cand. Sci. (Phys.–Math.),
Trapeznikov Institute of Control Sciences,
Russian Academy of Sciences, Moscow, Russia
✉ alexander.mazurov08@gmail.com