

МЕТОД ФУНКЦИОНАЛЬНОГО ДИАГНОСТИРОВАНИЯ ДИСКРЕТНО-СОБЫТИЙНЫХ СИСТЕМ НА ОСНОВЕ МОДЕЛИ НЕЧЕТКОГО КОНЕЧНОГО АВТОМАТА[#]

А. Е. Шумский*, А. Н. Жирабок**

***Дальневосточный федеральный университет, г. Владивосток

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

Аннотация. Рассматривается задача функционального диагностирования дискретно-событийных систем ответственного назначения, описываемых моделью нечеткого конечного автомата. Предлагается метод решения задачи, особенностью которого является использование математического аппарата нечеткой логики. Описываются операции нечеткой логики и вводится понятие детерминизатора нечеткого конечного автомата. Приводится схема диагностирования, позволяющая формировать структурированный вектор невязки. Схема содержит ряд каналов (по числу возможных дефектов в системе). В основе каждого канала лежит наблюдатель в виде детерминизатора специального нечеткого конечного автомата, одновременно учитывающего возможность как правильного перехода автомата при исправном функционировании системы, так и неправильного перехода, обусловливаемого дефектом в системе. Другой частью канала является блок принятия решения. Предлагаются способы синтеза наблюдателя и блока принятия решения. Особенности применения метода иллюстрируются на примере задачи мониторинга ошибок операторов ИТ-системы.

Ключевые слова: дискретно-событийные системы, нечеткая логика, нечеткие конечные автоматы, детерминизатор, функциональное диагностирование, ИТ-системы, мониторинг.

ВВЕДЕНИЕ

Удовлетворение жестких требований, предъявляемых к надежности и отказоустойчивости современных сложных систем ответственного назначения, требует осуществления оперативного диагностирования (обнаружения и поиска) дефектов, возникающих в процессе функционирования систем, в целях последующего парирования или устранения обнаруженных дефектов. В настоящей статье рассматривается задача функционального диагностирования дефектов в так называемых дискретно-событийных системах (ДСС). К ДСС относятся не только системы, естественным образом попадающие в этот класс (например, цифровые системы обработки информации и управления).

Многие системы, традиционно относящиеся к непрерывным (как то: физические, технические (в том числе человеко-машинные) и социально-экономические), могут рассматриваться как ДСС на верхнем уровне своей иерархии. Отличительные особенности ДСС состоят в следующем [1]:

- Дискретно-событийные системы являются дискретными по времени и значениям переменных состояния.
- Пространство состояний ДСС конечно; состояния могут иметь следующий смысл: ДСС простаивает, ДСС работает в определенном режиме, ДСС неисправна, ДСС восстанавливается и т. п.
- Функционирование ДСС определяется событиями, которые могут быть следствиями выполнения различных команд, например: «приступить к работе», «изменить режим работы», «осуществить диагностику ДСС», «приступить к восстановлению ДСС», «завершить восстановление ДСС», «завершить работу ДСС» и т. п.

[#] Работа выполнена при финансовой поддержке Минобрнауки России (проект FZNS-2023-0011).

• Дискретно-событийная система, как правило, ведет себя случайным образом, что связано с возможностью реализации разных переходов из одного состояния в различные состояния, инициируемых одним и тем же событием.

Обобщенная схема диагностирования и отказоустойчивого управления ДСС представлена на рис. 1. В соответствии с этой схемой контроллер генерирует внешние (по отношению к ДСС) события (входы ДСС или команды). В качестве ответа ДСС формирует внутренние (по отношению к ДСС) события, рассматриваемые как выходы ДСС. Контроллер отслеживает выходы ДСС, а также диагноз, определяемый системой диагностирования (СД), с целью формирования нового внешнего события (входа ДСС). В свою очередь, СД отслеживает как внешние, так и внутренние события для определения диагноза. Последний дает ответ на вопрос: исправна ли ДСС? Если ответ отрицательный, то делается дополнительное заключение о виде дефекта. Если дефект обнаружен и классифицирован, контроллер сообщает о новом событии, реакция на которое предполагает парирование дефекта (формирование последовательности команд, позволяющей исключить влияние дефекта на достижение цели) или устранение дефекта (ремонт ДСС).

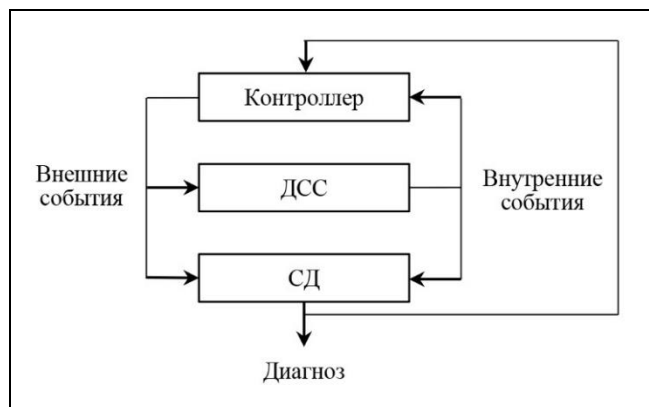


Рис. 1. Схема диагностирования и отказоустойчивого управления ДСС

Достаточно полный обзор существующих методов диагностирования ДСС можно найти в статьях [2, 3] и монографии [4]. В обзоре [3] отмечается, что для описания ДСС наиболее часто привлекаются следующие математические модели:

- детерминированные конечные автоматы,
- вероятностные конечные автоматы и марковские цепи,
- сети Петри.

Настоящая статья фокусируется на применении исключительно моделей конечного автомата (КА).

Использование подобных моделей для решения диагностических задач применительно к ДСС предполагает, что входы и выходы КА формируются как наблюдаемые события (внешние и внутренне соответственно), в то время как возникновение дефекта в ДСС рассматривается как непосредственно ненаблюдаемое внутреннее событие.

В тех случаях, когда детерминированная конечноавтоматная модель ДСС отсутствует, помимо модели в виде вероятностного КА также могут использоваться модели недетерминированного и нечеткого КА (НДКА и НЧКА соответственно). Применение недетерминированных и нечетких моделей (по сравнению с вероятностными) позволяет значительно сократить объем необходимых вычислений, что приводит к увеличению быстродействия процесса диагностирования. Модели НДКА и НЧКА учитывают ситуацию, когда для некоторого состояния и некоторого входа автомата могут быть реализованы переходы в различные последующие состояния. При этом, если в случае НДКА нельзя отдать приоритет возможности реализации того или иного перехода, то в случае НЧКА, пользуясь дополнительной (например, статистической) информацией, экспертными оценками, результатами обучения и т. п., можно говорить о степени уверенности в осуществлении каждого из возможных переходов.

Для диагностирования ДСС, описываемых моделью НДКА, в статьях [5] и [6] были представлены разработанные авторами методы, основанные на использовании соответственно парной алгебры разбиений и парной алгебры покрытий [7].

Целью настоящей статьи является разработка нового метода диагностирования на основе модели НЧКА. Для этого привлекаются математические конструкции нечеткой логики [8], а также понятия нечеткого автомата [9] и его детерминизатора [10].

Иллюстрация особенностей применения разработанного метода осуществляется на основе модели процесса управления изменениями в ИТ-системе, ранее рассмотренной в работах [5] и [6] применительно к задаче диагностирования НДКА.

1. РЕШАЕМЫЕ ЗАДАЧИ И СТРУКТУРА СТАТЬИ

1.1. Используемые модели

Пусть при исправной работе ДСС описывается НЧКА-моделью вида

$$A = (U, X, Y, \delta, \lambda, x(0)), \quad (1)$$

где $U = \{u_1, u_2, \dots, u_m\}$, $X = \{x_1, x_2, \dots, x_n\}$ и $Y = \{y_1, y_2, \dots, y_l\}$ – конечные множества входов,

состояний и выходов соответственно; $x(0) \in X$ – известное начальное состояние; $\delta: X \times U \rightarrow \mu(X)$ – нечеткая функция переходов; $\mu(X) \in \{\mu(x_i) \in [0, 1], 1 \leq i \leq n\}$ – нечеткое множество; $\lambda: X \rightarrow Y$ – функция выхода.

Для нечеткой функции переходов δ будем использовать матричное задание, при котором переходы, осуществляемые под воздействием входа u_k , описываются матрицей M^k размера $n \times n$, элемент которой $M_{i,j}^k \in [0, 1]$ характеризует степень уверенности в том, что автомат A из состояния $x_i \in X$ под воздействием входа $u_k \in U$ перейдет в состояние $x_j \in X$. Обозначим через $S(U)$ множество всех матриц $M^k, 1 \leq k \leq m$.

Если обоснованное задание степени уверенности в возможности осуществления переходов невозможно, предлагается поступать следующим образом: элемент матрицы $M_{i,j}^k$, соответствующий допустимому переходу, задается равным единице, в то время как элемент этой матрицы, соответствующий недопустимому переходу, принимается равным нулю. Тем самым осуществляется переход от НЧКА-модели ДСС к ее НДКА-модели.

Для задания функции выхода λ будем использовать матрицу L размера $l \times n$, элемент которой $L_{i,j} = 1$, если выход $y_i \in Y$ формируется автоматом A в состоянии $x_j \in X$, в противном случае $L_{i,j} = 0$. Обозначим через $S(Y)$ множество всех строк $\{L_i, 1 \leq i \leq l\}$ матрицы L .

Пусть $X_{i,k}, X_{i,k} \subseteq X$, – множество всех состояний, достижимых из состояния $x_i \in X$ под воздействием входа $u_k \in U$. Предположим, что дефект $f_s, 1 \leq s \leq N$, в ДСС на модели (1) может быть представлен искажением функции перехода δ таким, что вместо допустимого перехода из состояния x_i в состояние $x_j \in X_{i,k}$ под воздействием входа $u_k \in U$ реализуется недопустимый переход в состояние $x_t \notin X_{i,k}$. Этот факт будем отражать следующим образом: $f_s: (x_j \rightarrow x_t)_{i,k}$. При этом матрицу M^k ДСС с дефектом получим изменением значения элемента $M_{i,t}^k$ матрицы M^k исправной ДСС с нуля на единицу. Для упрощения даль-

нейшего изложения будем придерживаться гипотезы об однократных дефектах из заранее заданного списка $F = \{f_1, f_2, \dots, f_N\}$. Обозначим через A_s вспомогательный автомат, функция перехода δ_s которого получена изменением (способом, как было указано выше) функции перехода δ автомата A на переход, обусловливаемый дефектом f_s :

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

Модель (2) в силу особенностей построения матрицы M^k учитывает возможность «правильного» перехода при исправной работе ДСС, но также допускает и возможность «неправильного» перехода в состояние, обусловленное дефектом f_s в ДСС. Как следствие, можно говорить об определенном соответствии между поведением модели (2) и поведением как исправной ДСС, так и ДСС с дефектом. Способ оценки этого соответствия в терминах возможности (уверенности) предлагается в п. 3.2 и лежит в основе формирования структурированного вектора невязок.

1.2. Схема диагностирования

Для обнаружения и поиска дефектов предлагается использовать схему диагностирования, приведенную на рис. 2. Эта схема содержит N каналов (по числу возможных дефектов в ДСС), где каждый из каналов включает детерминированный КА A_s^d и блок принятия решения БПР_{*s*}. Синтез схемы диагностирования предполагает нахождение составляющих каждого канала. Необходимость решения этих задач определяет следующую структуру статьи.

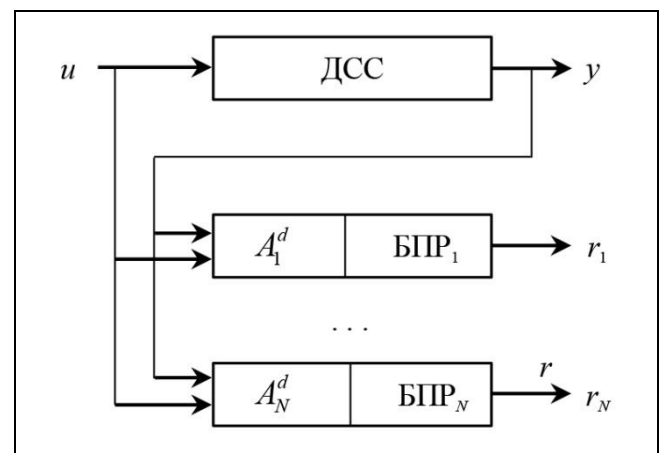


Рис. 2. Схема диагностирования ДСС



1.3. Структура статьи

Статья структурирована следующим образом. В § 2 приводятся те конструкции математического аппарата нечеткой логики, которые потребуются для получения основных результатов статьи. В § 3 решение задачи синтеза ДКА A_s^d , называемого наблюдателем НДКА A_s , сводится к модифицированной задаче нахождения детерминизатора [10] этого автомата. Далее в этом же параграфе предлагается правило функционирования БПР_s, $1 \leq s \leq N$, обеспечивающее свойство структурированности вектора невязки. Суть этого свойства состоит в следующем. Равенство нулю всех компонент вектора невязки означает отсутствие дефектов в ДСС. Равенство нулю только одной из компонент вектора невязки, в то время как остальные компоненты равны единице, соответствует случаю возникновения дефекта в ДСС, номер которого совпадает с номером нулевой компоненты вектора невязки. В § 4 рассматривается пример и приводятся результаты моделирования, иллюстрирующие особенности предложенного метода. В § 5 содержатся основные выводы по статье.

2. МАТЕМАТИЧЕСКИЕ КОНСТРУКЦИИ

2.1. Операции нечеткой логики

Напомним ряд операций нечеткой логики [8], играющих важную роль для дальнейшего изложения. Нечеткая матрица $B = \{B_{ij}\}$ – это матрица с элементами $B_{ij} \in [0, 1]$. Следовательно, введенные выше матрицы M^k , $1 \leq k \leq m$, – суть нечеткие матрицы. Пусть теперь B и C – нечеткие матрицы размера $a \times b$ и $b \times c$ соответственно. Произведение нечетких матриц определяется формулой [10]

$$(BC)_{ij} = \max_{h} \min_{j} (B_{ih} C_{hj}),$$

где B_{ih} и C_{hj} – соответствующие элементы матриц B и C , $1 \leq i \leq a$, $1 \leq h \leq b$, $1 \leq j \leq c$. Эта формула является обобщением известной формулы умножения матриц [11, с. 24]; она получается заменой произведения элементов матриц операцией нахождения минимума и суммы элементов – операцией нахождения максимума. Рассмотрим простой пример.

Пример 1. Пусть матрицы B и C имеют вид:

$$B = \begin{pmatrix} 0,1 & 0,9 \\ 0,7 & 0,2 \end{pmatrix}, C = \begin{pmatrix} 0,6 & 0,3 \\ 0,4 & 0,8 \end{pmatrix}.$$

После проведенных вычислений получим: $(BC)_{11} = \max(\min(0,1 \ 0,6), \min(0,9 \ 0,4)) = 0,4$, $(BC)_{12} = \max(\min(0,1 \ 0,3), \min(0,9 \ 0,8)) = 0,8$, $(BC)_{21} = \max(\min(0,7 \ 0,6), \min(0,2 \ 0,4)) = 0,6$, $(BC)_{22} = \max(\min(0,7 \ 0,3), \min(0,2 \ 0,8)) = 0,3$. В результате произведение нечетких матриц B и C принимает вид:

$$BC = \begin{pmatrix} 0,4 & 0,8 \\ 0,6 & 0,3 \end{pmatrix} \cdot \blacklozenge$$

2.2. Понятие детерминизатора нечеткого конечного автомата

Определение детерминизатора НЧКА [10] вводится следующим образом. Пусть E – некоторое множество нечетких векторов размера $1 \times n$, компоненты которых могут принимать значения на интервале $[0, 1]$. Пусть также $E^0, E^0 \subseteq E$, – множество всех единичных векторов размера $1 \times n$, называемых порождающими состояниями детерминизатора $D(A')$. Замыкание $[E]_\Sigma$ множества E относительно сигнатуры (множества допустимых операций) Σ есть множество всех векторов (включающее сами векторы множества E), которые можно получить, применяя операции из сигнатуры Σ к векторам из множества E . Для построения замыкания предлагается использовать следующую процедуру:

1. Положить $i = 0$.
2. Найти множество векторов $E^{i+1} = E^i \cup [E^i]_\Sigma$.
3. Если $E^{i+1} = E^i$, положить $[E]_\Sigma = E^i$. Конец.
4. Иначе положить $i = i + 1$ и перейти к п. 2.

Пусть $A' = (U, X, \delta)$ – нечеткий полуавтомат (т. е. автомат A без функции выхода), $S(U) = \{M^k, 1 \leq k \leq m\}$ – множество нечетких матриц перехода автомата A и сигнатура Σ включает все операции, предполагающие умножение вектора размерности $1 \times n$ справа на матрицу из множества $S(U)$. Детерминизатором нечеткого полуавтомата A' называется ДКА, описываемый тройкой

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

где $S(X) = [E]_{S(U)}$, а $\Delta: S(X) \times S(U) \rightarrow S(X)$ – функция перехода детерминизатора, определяемая соотношением

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

Для пояснения формулы (3) напомним, что матрица M^k описывает переход, активируемый входом u_k .

Пример 2. Пусть полуавтомат A' задается нечеткими матрицами перехода

$$M^1 = \begin{pmatrix} 0,4 & 0,6 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}, M^2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

Зададим порождающие состояния детерминизатора: $\mu_1 = (1 \ 0 \ 0)$, $\mu_2 = (0 \ 1 \ 0)$, $\mu_3 = (0 \ 0 \ 1)$. Осуществляя необходимые вычисления согласно формуле (3), найдем векторы μ_4 , μ_5 и μ_6 замыкания $[E]_{S(U)}: \mu_4 = \mu_1 \cdot M^1 = (0,4 \ 0,6 \ 0)$, $\mu_5 = \mu_4 \cdot M^1 = (0,4 \ 0,4 \ 0,6)$, $\mu_6 = \mu_4 \cdot M^2 = (0,6 \ 0 \ 0,4)$. Рассуждая аналогичным образом, получим: $\mu_7 = (0,4 \ 0,6 \ 0,4)$, $\mu_8 = (0,6 \ 0,4 \ 0,4)$, $\mu_9 = (0 \ 0,4 \ 0,6)$, $\mu_{10} = (0 \ 0 \ 0,4)$, $\mu_{11} = (0 \ 0,4 \ 0)$, $\mu_{12} = (0,4 \ 0 \ 0)$, $\mu_{13} = (0,4 \ 0,4 \ 0)$, $\mu_{14} = (0,4 \ 0,4 \ 0,4)$, $\mu_{15} = (0,4 \ 0 \ 0,4)$, $\mu_{16} = (0 \ 0,4 \ 0,4)$. Окончательно, принимая $M(X) = \{\mu_i, 1 \leq i \leq 16\}$, построим таблицу переходов детерминизатора $D(A')$ (табл. 1). В этой таблице, например, указаны переходы из состояния μ_9 в состояние μ_{10} под воздействием входа u_1 и из состояния μ_9 в состояние μ_4 под воздействием входа u_2 , поскольку $\mu_9 \cdot M^1 = \mu_{10}$ и $\mu_9 \cdot M^2 = \mu_4$.

Заметим, что число состояний детерминизатора $D(A')$ существенно превосходит число состояний исходного полуавтомата A' . Это связано с тем, что при построении детерминизатора не происходит потери информации о степени уверенности в осуществлении каждого из возможных переходов, поскольку каждое состояние детерминизатора –

это вектор, компоненты которого равны возможностям перехода ДСС в соответствующие свои состояния в определенный момент времени. ♦

Для получения верхней граничной оценки числа состояний детерминизатора учтем, что компоненты векторов из множества $S(X)$ могут принимать только те значения, которые содержатся в матрицах из множества $S(U)$. Исключая из рассмотрения нулевой вектор, получим формулу:

$$\#S(X) \leq q^n - 1, \quad (4)$$

где символ $\#$ обозначает мощность множества; q – число различных значений элементов нечетких матриц из множества $S(U)$; n , как и ранее, – число состояний НЧКА. Так, для системы (3) имеем: $q = 4$ (множество различных значений элементов матриц M^1 и M^2 включает числа $\{0; 0,4; 0,6; 1\}$), $n = 3$ и, как следствие, $\#S(X) \leq 4^3 - 1 = 63$.

Пример 3. Описанный выше подход также может быть применен и для детерминизации НДКА. Проиллюстрируем это на примере модели НДКА, полученной из НЧКА-модели (3) посредством задания единичных значений степени уверенности для всех допустимых переходов:

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

Отметим, что рассматриваемый подход и в случае НДКА не гарантирует получение детерминизатора, размерность которого меньше размерности исходного автомата. Действительно, так как для НДКА $q = 2$, то согласно выражению (4) число состояний детерминизатора будет определяться формулой

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

Таблица 1

Таблица переходов детерминизатора $D(A)$

	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}



В частности, для НДКА (5) имеем $\#S(X) \leq 2^3 - 1 = 7$.
Опуская промежуточные выкладки, приведем таблицу переходов детерминизатора НДКА (5).

Таблица 2

Таблица переходов детерминизатора НДКА (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

В табл. 2 состояния детерминизатора таковы:
 $\mu_1 = (1 \ 0 \ 0)$, $\mu_2 = (0 \ 1 \ 0)$, $\mu_3 = (0 \ 0 \ 1)$, $\mu_4 =$
 $= (1 \ 1 \ 0)$, $\mu_5 = (1 \ 1 \ 1)$, $\mu_6 = (1 \ 0 \ 1)$, $\mu_7 =$
 $= (0 \ 1 \ 1)$. ♦

Заметим, что алгебраический подход к детерминизации НДКА (алгебра разбиений и алгебра покрытий), описанный соответственно в работах [5] и [6], гарантирует нахождение ДКА, размерность которого не превышает размерности исходного автомата. Это обуславливается тем, что в рамках алгебраического подхода не ставится задача сохранения информации о степени уверенности в осуществлении каждого из возможных переходов. Рассматриваемый нечеткий подход позволяет использовать при диагностировании дополнительную полезную информацию о степени уверенности в осуществлении каждого из возможных переходов, что предоставляет потенциальную возможность для увеличения (при необходимости) глубины поиска дефектов. Плата за это – значительная размерность таблицы переходов детерминизатора.

Основное отличие описания ДКА A_s^d , $1 \leq s \leq N$, из схемы диагностирования, приведенной на рис. 2, от описания детерминизатора $D(A_s)$ соответствующего полуавтомата A_s' состоит в том, что функция переходов автомата A_s^d дополнительно зависит от выходов исходного автомата. Это позволяет использовать дополнительную информацию для корректировки поведения и снижения размерности автомата A_s^d . Проводя аналогию с наблюдателями непрерывной динамической системы, ДКА A_s^d назовем наблюдателем НЧКА A_s .

3. СИНТЕЗ КАНАЛА ДИАГНОСТИРОВАНИЯ

3.1. Синтез наблюдателя A_s^d

Дальнейшее изложение привязано к каналу схемы диагностирования, ориентированному на поиск дефекта f_s . Рассмотрим сначала способ нахождения функции перехода δ_s^d наблюдателя A_s^d , основанный на небольшой модификации соотношения (3). Для векторов $\mu_i \in S(X)$ и $L_j \in S(Y)$ введем следующее обозначение:

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

Функцию перехода $\Delta: S(X) \times S(Y) \times S(U) \rightarrow S(X)$ ДКА A_s^d зададим следующим образом:

$$\Delta(\mu_i, y_j, u_k) = \langle \mu_i, L_j \rangle \cdot M^k, \quad (7)$$

$$\mu_i \in S(X), L_j \in S(Y), M^k \in S(U).$$

Использование $\langle \mu_i, L_j \rangle$ в формуле (7) вместо вектора μ_i в выражении для функции перехода (3) детерминизатора позволяет уточнить значения функций принадлежности компонент вектора $x \in X$ соответствующим нечетким множествам до осуществления перехода.

Пример 4. Пусть НЧКА при отсутствии дефектов описывается матрицами M^1 и M^2 из примера 2. Пусть также функция выхода этого автомата задается матрицей

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

Ошибки переходов автомата являются следствием двух дефектов: $f_1: (x_3 \rightarrow x_1)_{1,2}$ и $f_2: (x_3 \rightarrow x_2)_{1,2}$. Матрицы M_1^2 и M_2^2 автомата, включающие дополнительный ошибочный переход, обусловленный дефектом (нижний индекс при матрице указывает на номер дефекта), имеют вид

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

Матрица M^1 не подвержена воздействию дефекта и сохраняет свой вид.

Проиллюстрируем синтез наблюдателя A_1^d (в этом случае используется только матрица M_1^2). Как и в предыдущем примере, примем порождающие состояния $\mu_1 = (1 \ 0 \ 0)$, $\mu_2 = (0 \ 1 \ 0)$, $\mu_3 = (0 \ 0 \ 1)$. Осуществляя вычисления согласно правой части соотношения (7), получим:

$$\langle \mu_1, L_1 \rangle \cdot M^1 = (\min(1, 1) \ \min(0, 0) \ \min(0, 0)) \times \begin{pmatrix} 0,4 & 0,6 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} = (0,4 \ 0,6 \ 0) = \mu_4.$$

Аналогично можно найти $\mu_5 = (1 \ 0 \ 1)$, $\mu_6 = (0,4 \ 0,4 \ 0)$, $\mu_7 = (0 \ 0 \ 0,6)$, $\mu_8 = (0,4 \ 0 \ 0,4)$, $\mu_9 = (0,6 \ 0 \ 0)$, $\mu_{10} = (0 \ 0 \ 0,4)$, $\mu_{11} = (0,4 \ 0 \ 0)$, $\mu_{12} = (0 \ 0,6 \ 0)$, $\mu_{13} = (0 \ 0,4 \ 0)$, $\mu_{14} = (0,6 \ 0 \ 0,6)$ и построить таблицу переходов (табл. 3) наблюдателя A_1^d .

В табл. 3 прочерком обозначены переходы, соответствующие несовместимым значениям состояния наблюдателя и выхода ДДС. Например, состояние наблюдателя μ_2 допускает только выход y_2 ДСС, что непосредственно следует из записи вектора μ_2 и матрицы L . Следовательно, сочетание μ_2 и y_1 невозможно при исправной работе ДСС.

Сравнивая размерность детерминизатора $D(A')$ (см. табл. 1) и размерность наблюдателя A_1^d (см. табл. 3), несложно видеть, что размерность наблюдателя не превышает размерность детерминизатора. Однако при этом размерность наблюдателя на практике может превышать размерность исходного НЧКА. ♦

Казалось бы, последний факт должен накладывать ограничения на возможность практической реализации процесса диагностирования ДСС на основе модели НЧКА, обуславливаемые значительной размерностью реальных систем. Эти ограничения представляются менее выраженными при использовании алгебраического подхода [5, 6]. Однако в действительности это не так. Дело в том, что применение алгебраического подхода ориен-

тировано на нахождение табличного описания функции переходов наблюдателей (ДКА A_i^d , $1 \leq i \leq N$) на основе табличного описания исходной конечноавтоматной модели ДСС. Предлагаемый же подход ориентирован на задание функции переходов в компактном аналитическом виде, что делает его малочувствительным к росту размерности исходной модели ДСС. Более того, в силу отмеченной особенности предлагаемый подход представляется предпочтительным и в случае синтеза схем диагностирования ДСС на основе ДКА-и НДКА-моделей. При его применении преодолевается так называемое «проклятие размерности», характерное для всех методов, ориентированных на табличное (или в виде графов) описание конечноавтоматной модели ДСС.

3.2. Построение блока принятия решения канала диагностирования

Предположим, что при диагностировании ДСС одновременно генерируются выход ДСС y_j и состояние μ_i наблюдателя A_s^d . Обозначим через σ_s значение максимальной компоненты вектора $\langle \mu_i, L_j \rangle$. Это значение будем рассматривать как степень уверенности в том, что поведение ДСС соответствует поведению наблюдателя A_s^d . Сказанное является основой для задания отношения Ψ_s :

$$(\mu_i, y_j) \in \Psi_s \Leftrightarrow \sigma_s \neq 0. \quad (8)$$

Пусть теперь отношение Ψ_s построено согласно выражению (8) и его проверка для конкретной пары y_j и μ_i показывает, что оно не выполняется (т. е. имеет место равенство $\sigma_s = 0$). Это будет свидетельствовать о том, что в ДСС произошла ошибка (дефект) f_k , $k \neq s$. Формируемое при этом значение невязки $r_s = 1$. В противном случае ($\sigma_s \neq 0$) формируется $r_s = 0$. Для того, чтобы сде-

Таблица 3

Таблица переходов наблюдателя 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}



лать однозначное заключение о виде дефекта в ДСС на основе структурированного вектора невязок, только одна его компонента, номер которой соответствует номеру дефекта, должна сохранять нулевое значение. Если это не так (т. е. сформировано несколько нулевых невязок), то требуются дополнительные проверки, например, с использованием дополнительных измерений и (или) специальных тестов [12]. При этом порядок проведения проверок целесообразно осуществлять в направлении дефектов, номера которых следуют в порядке уменьшения степени уверенности из списка $\{\sigma_1, \sigma_2, \dots, \sigma_N\}$.

4. ИЛЛЮСТРАТИВНЫЙ ПРИМЕР

В работах [5, 6] приведено подробное описание процесса управления изменениями ИТ-системы ([5], рис. 1) и рассмотрен способ получения его ДКА- и НДКА-моделей. Отметим, что процесс управления изменениями является одним из важнейших для ИТ-систем: он отвечает за управление жизненным циклом всех изменений и способствует реализации полезных изменений с минимальным прерыванием предоставления ИТ-услуг. Субъектами процесса являются:

- инициатор – представитель департамента ИТ, осуществляющий первичную обработку, назначение и контроль над ходом выполнения изменений;
- исполнитель – инженер, производящий изменения в элементах конфигурации или координирующий работы подрядчика по этим изменениям;
- консультативный комитет по изменениям (КИ) – консультативный орган, регулярно собирающийся для оценки и планирования изменений;
- менеджер процесса – представитель департамента ИТ, осуществляющий контроль процесса управления изменениями и формирующий предложения по его улучшению.

В этих же работах приведены таблицы переходов и выходов ДКА- и НДКА-моделей процесса. Модель ДКА описывает действия субъектов процесса в полном соответствии с предписанным регламентом. Необходимость использования исходной модели в виде НДКА обуславливается тем, что на практике допустимы некоторые (некритичные) отклонения действия субъектов от заданного регламента, направленные на уточнение самого регламента либо уточнение их знаний о регламенте. В частности, рассматривались следующие ситуации:

- менеджер процесса управления изменениями отправляет результат проверки выполненного задания на повторное согласование в комитет КИ;
- менеджер процесса управления изменениями отправляет полученное согласованное задание обратно инициатору;
- исполнитель отправляет полученное согласованное задание обратно инициатору;

– исполнитель отправляет полученное и согласованное задание обратно на согласование в комитет КИ.

В качестве входов модели рассматриваются следующие внешние события: u_1 – завершение формирования плана работ; u_2 – утверждение плана КИ; u_3 – передача не утвержденного КИ плана на доработку; u_4 – передача задания и плана работ исполнителю; u_5 – закрытие работ; u_6 – занесение внесенных изменений в библиотеку состояния ИТ-системы. В качестве состояний модели рассматриваются следующие этапы регламента: x_1 – формирование задания и плана реализации; x_2 – согласование задания и плана работ КИ; x_3 – согласование задания и плана работ менеджером процесса; x_4 – выполнение работ исполнителем; x_5 – проверка менеджером процесса выполненных работ; x_6 – работы завершены. В качестве выходов используется доступная информация о некоторых этапах выполнения регламента (функция выхода приводится ниже).

Описывающая безошибочную работу участников процесса (инициатор, исполнитель и менеджер) НЧКА-модель процесса управления изменениями ИТ-системы может быть получена на основе НДКА-модели ([6], табл. 3) и имеющейся статистической информации о возможных действиях участников при выполнении регламента:

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

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

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

$$M^4 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0,1 & 0,1 & 0 & 0,8 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

$$M^5 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0,2 & 0,1 & 0 & 0 & 0,7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

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

Все вышеприведенные матрицы имеют размерность 6×6 . Функция выхода описывается матрицей

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

Как и в работе [6], будем рассматривать ошибки инициатора, исполнителя и менеджера процесса соответственно: $f_1 : (x_2 \rightarrow x_3)_{(x_1, u_1)}$, $f_2 : (x_5 \rightarrow x_6)_{(x_4, u_5)}$, $f_3 : (x_3 \rightarrow x_4)_{(x_2, u_2)}$. Этим ошибкам соответствуют матрицы

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

$$M_2^5 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0,2 & 0,1 & 0 & 0 & 0,7 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix},$$

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

Все остальные матрицы свои значения сохраняют.

В приведенных ниже таблицах представлены результаты моделирования, полученные для различных допустимых результатов работы субъектов процесса управления изменениями, а также ошибок инициатора, исполнителя и менеджера процесса. Генерация невязок и вспомогательных переменных осуществлялась на основе соотношений (6)–(8) с использованием приведенных выше матриц. Для первого, второго и третьего каналов используются матрицы M_1^1 , M_2^2 , M_3^3 , M_4^4 , M_5^5 , M_6^6 ; M_1^1 , M_2^2 , M_3^3 , M_4^4 , M_5^5 , M_6^6 и M_1^1 , M_3^3 , M_4^4 , M_5^5 , M_6^6 соответственно. При моделировании авторы ограничились следующими сценариями.

Сценарий 1. Регламент процесса реализуется без каких-либо отклонений (табл. 4).

Сценарий 2. Регламент процесса выполняется с допустимым отклонением, обусловленным передачей менеджером задачи формирования задания и плана работ для повторного согласования в КИ (табл. 5).

Сценарий 3. Регламент процесса выполняется с ошибками, допускаемыми соответственно инициатором, исполнителем и менеджером процесса (табл. 6).

Таблица 4

Сценарий 1

Характеристики		Начальное значение	Вход				
			u_1	u_2	u_4	u_5	u_6
Система	Состояние x	x_1	x_2	x_3	x_4	x_5	x_6
	Выход y	y_1	y_2	y_3	y_4	y_2	y_5
Канал 1	Состояние μ	μ_{10}	μ_{11}	μ_{12}	μ_{14}	μ_{15}	μ_{16}
	Уверенность σ_1	1	1	1	0,8	0,7	0,7
	Невязка r_1	0	0	0	0	0	0
Канал 2	Состояние μ	μ_{20}	μ_{21}	μ_{22}	μ_{24}	μ_{25}	μ_{26}
	Уверенность σ_2	1	1	1	0,8	0,7	0,7
	Невязка r_2	0	0	0	0	0	0
Канал 3	Состояние μ	μ_{30}	μ_{31}	μ_{32}	μ_{34}	μ_{35}	μ_{36}
	Уверенность σ_3	1	1	1	0,8	0,7	0,7
	Невязка r_3	0	0	0	0	0	0



Таблица 5

Сценарий 2

Характеристики		Начальное значение	Вход						
			u_1	u_2	u_4	u_2	u_4	u_5	u_6
Система	Состояние x	x_1	x_2	x_3	x_2	x_3	x_4	x_5	x_6
	Выход y	y_1	y_2	y_3	y_2	y_3	y_4	y_2	y_5
Канал 1	Состояние μ	μ_{10}	μ_{11}	μ_{12}	μ_{14}	μ_{12}^*	μ_{14}^*	μ_{15}^*	μ_{16}^*
	Уверенность σ_1	1	1	1	0,1	0,1	0,1	0,1	0,1
	Невязка r_1	0	0	0	0	0	0	0	0
Канал 2	Состояние μ	μ_{20}	μ_{21}	μ_{22}	μ_{24}	μ_{22}^*	μ_{24}^*	μ_{25}^*	μ_{26}^*
	Уверенность σ_2	1	1	1	0,1	0,1	0,1	0,1	0,1
	Невязка r_2	0	0	0	0	0	0	0	0
Канал 3	Состояние μ	μ_{30}	μ_{31}	μ_{32}	μ_{34}	μ_{32}^*	μ_{34}^*	μ_{35}^*	μ_{36}^*
	Уверенность σ_3	1	1	1	0,1	0,1	0,1	0,1	0,1
	Невязка r_3	0	0	0	0	0	0	0	0

Таблица 6

Сценарий 3

Характеристики		Начальное значение	f_1	f_2				f_3	
			Вход						
			u_1	u_1	u_2	u_4	u_5	u_1	u_2
Система	Состояние x	x_1	x_3	x_2	x_3	x_4	x_6	x_2	x_4
	Выход y	y_1	y_3	y_2	y_3	y_4	y_5	y_2	y_4
Канал 1	Состояние μ	μ_{10}	μ_{11}	μ_{11}	μ_{12}	μ_{14}	μ_{15}	μ_{11}	μ_{12}
	Уверенность σ_1	1	1	1	1	0,8	0	1	0
	Невязка r_1	0	0	0	0	0	1	0	1
Канал 2	Состояние μ	μ_{20}	μ_{21}	μ_{21}	μ_{22}	μ_{24}	μ_{25}	μ_{21}	μ_{22}
	Уверенность σ_2	1	0	1	1	0,8	0,8	1	0
	Невязка r_2	0	1	0	0	0	0	0	1
Канал 3	Состояние μ	μ_{30}	μ_{31}	μ_{31}	μ_{32}	μ_{34}	μ_{35}	μ_{31}	μ_{32}
	Уверенность σ_3	1	0	1	1	0,8	0	1	1
	Невязка r_3	0	1	0	0	0	1	0	0

Отметим, что реализация различных сценариев требует выполнения каждого из этапов регламента разное число раз, что нашло своё отражение в таблицах. Состояния наблюдателей каналов диагностирования, фигурирующие в таблицах, имеют следующие значения: $\mu_{10} = \mu_{20} = \mu_{30} = (1\ 0\ 0\ 0\ 0\ 0)$, $\mu_{11} = (0\ 1\ 1\ 0\ 0\ 0)$, $\mu_{21} = \mu_{31} = (0\ 1\ 0\ 0\ 0\ 0)$, $\mu_{12} = \mu_{22} = (0\ 0\ 1\ 0\ 0\ 0)$, $\mu_{32} = (0\ 0\ 1\ 1\ 0\ 0)$, $\mu_{14} = \mu_{24} = \mu_{34} = (0,1\ 0,1\ 0\ 0,8\ 0\ 0)$, $\mu_{15} = \mu_{35} = (0,2\ 0,1\ 0\ 0\ 0\ 0,7\ 0)$, $\mu_{25} = (0,2\ 0,1\ 0\ 0\ 0\ 0,7\ 0,8)$,

$\mu_{16} = \mu_{26} = \mu_{36} = (0\ 0\ 0\ 0\ 0\ 0,7)$. Состояния наблюдателей каналов, фигурирующие в табл. 5, имеют следующие значения: $\mu_{12}^* = \mu_{22}^* = (0\ 0\ 0,1\ 0\ 0\ 0)$, $\mu_{32}^* = (0\ 0\ 0,1\ 0,1\ 0\ 0)$, $\mu_{14}^* = \mu_{24}^* = \mu_{34}^* = (0,1\ 0,1\ 0\ 0,1\ 0\ 0)$, $\mu_{15}^* = \mu_{35}^* = (0,1\ 0,1\ 0\ 0\ 0,1\ 0)$, $\mu_{25}^* = (0,1\ 0,1\ 0\ 0\ 0,1\ 0,1)$, $\mu_{16}^* = \mu_{26}^* = \mu_{36}^* = (0\ 0\ 0\ 0\ 0\ 0,1)$. Нижние индексы при состояниях имеют следующий смысл: первый индекс соответствует номеру канала, второй индекс соответ-

ствуется номеру входа, под воздействием которого осуществляется переход в это состояние.

Верхний индекс * вводится для удобства обозначения вновь появляющихся состояний при реализации сценария 2, генерируемых при тех же входах, что и в случае реализации сценариев 1 и 3; это связано с тем, что реализация сценария 2 требует выполнения большего числа этапов регламента.

Из табл. 4 и 5 следует, что как при полном соблюдении регламента, так и при допустимом отклонении от регламента, обусловливаемым действием менеджера процесса, на выходах каналов формируются нулевые значения невязок. В тоже время табл. 6 показывает, что при возникновении ошибок в действиях инициатора, исполнителя и менеджера процесса формируется структурированный вектор невязок, позволяющий сделать однозначный вывод о виде ошибки в момент ее проявления.

ЗАКЛЮЧЕНИЕ

В статье предложен метод диагностирования ДСС, описываемых моделью нечеткого конечного автомата. Использование подобной модели (по сравнению с недетерминированной моделью) может положительно сказаться на возможности достижения требуемой глубины диагностирования. Действительно, пусть сформированный вектор невязок не позволяет сделать однозначное заключение о виде дефекта (вектор содержит несколько нулевых компонент). В этом случае для локализации дефекта может потребоваться проведение ряда дополнительных проверок. С целью сокращения числа таких проверок их следует осуществлять в порядке, соответствующем уменьшению степени уверенности σ_s , $1 \leq s \leq N$.

Очевидным образом метод может быть распространен на случаи использования более простых моделей в виде детерминированных и недетерминированных конечных автоматов. Отличительной особенностью метода является то, что для его реализации не требуется предварительное нахождение табличного описания средств диагностирования; все вычисления осуществляются непосредственно в процессе диагностирования с использованием компактных аналитических соотношений. Тем самым преодолевается проблема «проклятия размерности», характерная для методов, ориентированных на табличное (или в виде графов) описание конечноавтоматной модели ДСС. Последнее позволяет практически снять ранее существовавшее ограничение на допустимую размерность модели

диагностируемой ДСС. Дальнейшее развитие предложенного метода предполагается осуществить на случай задачи анализа диагностируемости и верификации моделей ДСС с большим числом состояний (см. работу [13]).

ЛИТЕРАТУРА

1. Wonham, W.M., Kai C., Rudle, K. Supervisory Control of Discrete-Event Systems: A Brief History 1980-2015 // Proc. of 20th IFAC Congress. – Toulouse, France, 2017. – P. 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. – P. 105–124.
3. Zaytoon, J., Lafortune, S. Overview of Fault Diagnosis Methods for Discrete Event Systems // Annual Reviews in Control. – 2013. – Vol. 37, no. 2. – P. 308–320.
4. Cassandras, Ch., Lafortune, S. Introduction to Discrete Event Systems. Second edition. – New York: Springer Science+Business Media, LLC, 2008. – 770 p.
5. Журабок А. Н., Калинина Н. А., Шумский А. Е. Метод мониторинга поведения человека-оператора в человеко-машинных системах // Известия РАН. Теория и системы управления. – 2018. – № 3. – С. 98–107. [Zhirabok, A.N., Kalinina, N.A., 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. – P. 443–452.]
6. Журабок А. Н., Калинина Н. А., Шумский А. Е. Метод функционального диагностирования недетерминированных конечных автоматов // Известия РАН. Теория и системы управления. – 2020. – № 4. – С. 62–72. [Zhirabok, A.N., Kalinina, N.A., 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. – P. 565–574.]
7. Hartmanis, J., Stearns, R. The Algebraic Structure Theory of Sequential Machines. – New York: Prentice-Hall Inc., 1966. – 211 p.
8. Заде Л. Понятие лингвистической переменной и ее применение к принятию приближенных решений. – М: Мир, 1976. – 165 с. [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., 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. – P. 215–223.
10. Ульзутуев И.Е., Максимов А.А. О свойствах решёток подавтоматов нечётких полуавтоматов и их детерминизаторов // Вестник СГТУ. – 2015. – № 2 (79). – С. 117–126. [Ulyutuev, I.E., Maximov, A.A. The Properties of Subautomata Lattices of Fuzzy Semiautomata and Their Determinizers // Vestnik of Saratov State Technical University. – 2015. – No. 2 (79). – P. 117–126. (In Russian)]
11. Воеводин В.В., Кузнецов Ю.А. Матрицы и вычисления. – М.: Наука, 1984. – 320 с. [Voevodin, V.V., Kuznetsov, U.A. Matrices and computations. – Moscow: Nauka, 1984. – 320 p. (In Russian)]
12. Грузликов А.М., Колесов Н.В. Дискретно-событийная диагностическая модель распределенной вычислительной си-



- стемы. Независимые цепи // Автоматика и телемеханика. – 2016. – № 10. – С. 139–154. [Gruzlikov, A.M., Kolesov, N.V. Discrete-Event Diagnostic Model for a Distributed Computational System. Independent Chains // Automation and Remote Control. – 2017. – Vol. 77, no. 10. – P. 1805–1817.]
13. Tuxi, T.M., Carvalho, L.K., Nunes, E.V.L., Cunha, A.E. Diagnosability Verification Using LTL Model Checking // Discrete Event Dynamic Systems. – 2022 – Vol. 32, no. 3. – P. 399–433.

Статья представлена к публикации членом редколлегии
С.А. Красновой.

Поступила в редакцию 04.03.2025,
после доработки 17.04.2025.
Принята к публикации 29.04.2025.

Шумский Алексей Евгеньевич – д-р техн. наук,
ORCID iD: <https://orcid.org/0000-0002-5429-7482>
✉ a.e.shumsky@yandex.com

Жиравок Алексей Нилович – д-р техн. наук,
ORCID iD: <https://orcid.org/0000-0001-5927-7117>
✉ zhirabok@mail.ru

Дальневосточный федеральный университет, г. Владивосток

© 2025 г. Шумский А. Е., Жиравок А. Н.



Эта статья доступна по лицензии Creative Commons
«Attribution» («Атрибуция») 4.0 Всемирная.

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.

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