

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ОТКАЗОВ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

Е.В. Ларкин¹, А.В. Богомолов², А.Н. Привалов³

¹Тульский государственный университет, г. Тула, Российская Федерация

²Федеральный исследовательский центр «Информатика и управление» РАН, г. Москва, Российская Федерация

³Тульский государственный педагогический университет имени Л.Н. Толстого, г. Тула, Российская Федерация

В статье представлены результаты исследования причин возникновения отказов программного обеспечения в аспекте их математического моделирования. Предложен метод линеаризации структуры алгоритма, описанного в виде полумарковского процесса, основанный на многократном выполнении операции конкатенации над матрицей смежности. Разработана методика расчета параметров линеаризованной структурно-параметрической математической модели по параметрам алгоритма. Определен перечень структурных и параметрических ошибок, допускаемых при разработке программного обеспечения, и получены оценки вероятностей появления отказных и безотказных реализаций. Разработана полумарковская модель безотказной и отказной работы программного обеспечения. Установлено, что распределение вероятностей отказов по количеству отказных реализаций при заданном количестве реализаций описывается биномиальным законом, а при переходе во временную область интервал между отказами описывается экспоненциальным законом. Показано, что при коррекции программного обеспечения за счет исправления ошибок при его эксплуатации сокращается область отказных реализаций алгоритма, обеспечивая повышение надежности программного обеспечения.

Ключевые слова: отказ программного обеспечения; математическая модель отказа; полумарковский процесс; надежность программного обеспечения; полумарковская модель работы программного обеспечения; линеаризация алгоритма; интерпретация алгоритма.

Введение

Переход промышленности на новые цифровые технологии и применение систем искусственного интеллекта в управленческой деятельности актуализирует проблему надежности аппаратно-программных комплексов, используемых для обработки информации и принятия решений [1–3]. Отказы комплексов подобного типа определяются надежностью аппаратной платформы и установленного на ней программного обеспечения [4]. Причинами частичной или полной потери работоспособности аппаратно-программных комплексов являются естественные физические процессы, протекающие во времени: брак, допущенный при проектировании и изготовлении, несоблюдение условий эксплуатации, старение элементной базы и т. п. Алгоритмическое и программное обеспечение (ПО) представляет собой своего рода модель управления обработкой данных, которая разрабатывается человеком. Именно поэтому причиной отказов ПО являются исключительно ошибки, допущенные разработчиками и проявляющиеся при запусках программ во время обработки реальных данных [5]. Несмотря на различие механизмов генерации потоков отказов программной и аппаратной составляющими, при оценивании надежности аппаратно-программных комплексов должна быть построена единая модель [6–8], учитывающая обе компоненты потока – следовательно, подход к математическому моделированию отказов должен быть единым как для аппаратуры, так и для ПО.

Подход к моделированию работы ПО и отказов ПО связан с построением структурно-параметрической модели, в которой процесс функционирования сводится к последовательной смене состояний. При моделировании работы ПО под состоянием понимается собственно процедура интерпретации оператора алгоритма аппаратными средствами [9]. При моделировании отказов ПО под состоянием понимается отказный или безотказный результат обработки данных программой [10, 11]. В обоих случаях при моделировании используется теория марковских процессов [12–14], а, следовательно, модель отказов ПО может быть получена из модели обработки данных программными средствами с помощью формального структурно-параметрического преобразования.

Отсутствие методик преобразования моделей работы ПО в модели его отказов не позволяет производить корректировку программного обеспечения на этапе его разработки, что неоправданно увеличивает сроки разработки и тестирования программ. Таким образом, решение задачи преобразования полумарковской модели интерпретации алгоритма в марковскую модель генерации отказов ПО является актуальной.

1. Линеаризация модели обработки данных

Одним из широко применяемых подходов к моделированию процесса интерпретации алгоритма аппаратно-программных комплексов является формирование полумарковского процесса

$$\mu = \{A, \mathbf{r}, \mathbf{h}(t)\}, \quad (1)$$

где $A = \{a_0, a_1, \dots, a_i, \dots, a_j, a_{j+1}\}$ – множество состояний, совпадающее с множеством операторов; $\mathbf{r} = [r_{i,j}]$ – матрица смежности, описывающая структуру связей между состояниями полумарковского процесса, совпадающую со структурой связей между операторами алгоритма; $\mathbf{h}(t) = \mathbf{p} \otimes \mathbf{f}(t)$ – полумарковская матрица; $\mathbf{f}(t) = [f_{i,j}(t)]$ – матрица плотностей распределения времени выполнения операторов алгоритма; $\mathbf{p} = [p_{i,j}]$ – стохастическая матрица.

Дуга $r_{i,j}$ графа, отображающего структуру алгоритма (полумарковского процесса), – рис. 1 – представляет собой путь $r_{i,j} = (a_i, a_j)$ единичной длины, ведущий из оператора (состояния полумарковского процесса) a_i в оператор (состояние) a_j . Если такого пути не существует, то $r_{i,j} = \emptyset$, где $\emptyset$ – пустой элемент.

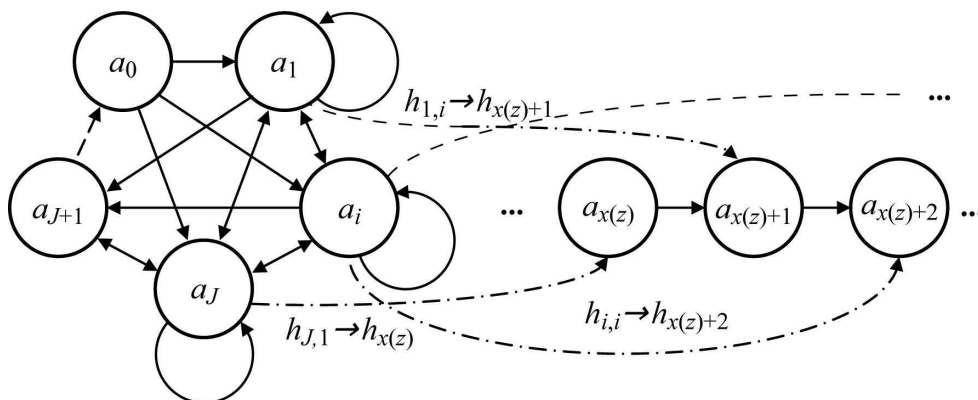


Рис. 1. Структура алгоритма

Из множества операторов (состояний) A могут быть выделены неисполнимый оператор «Начало» (стартовое состояние полумарковского процесса) a_0 и неисполнимый оператор «Конец» (поглощающее состояние) a_{j+1} (рис. 1). В матрице $\mathbf{r}$ столбец,

описывающий стартовое состояние, состоит из пустых элементов, т. е.

$$(r_{0,0}, \dots, r_{i,0}, \dots, r_{J+1,0})^\theta = (\emptyset, \dots, \emptyset, \dots, \emptyset)^\theta,$$

где θ – операция транспонирования.

Строка, описывающая поглощающее состояние, также состоит из пустых элементов, т. е. $(r_{J+1,0}, \dots, r_{J+1,i}, \dots, r_{J+1,J+1}) = (\emptyset, \dots, \emptyset, \dots, \emptyset)$. Все оставшиеся элементы могут быть как пустыми, так и непустыми. Дуга, изображенная на рис. 1 в виде пунктирной стрелки, замыкает обратную связь и моделирует перезапуск программы. При замкнутой обратной связи $(r_{J+1,0}, \dots, r_{J+1,i}, \dots, r_{J+1,J+1}) = [(a_{J+1}, 0), \emptyset, \dots, \emptyset, \dots, \emptyset]$. В графе, моделирующем правильно спроектированный алгоритм, для каждого оператора a_i существует хотя бы один путь $a_0 \rightarrow a_i$ и хотя бы один путь $a_i \rightarrow a_{J+1}$ (свойство достижимости операторов/состояний).

Назовем n -й путь $(a_0 \rightarrow a_{j+1})$ n -й траекторией реализации алгоритма (полумарковского процесса), которая определяется значениями вектора обрабатываемых данных и семантикой логических условий по выбору продолжения траектории реализации в местах ветвления алгоритма. В правильно спроектированном алгоритме каждый оператор $a_i \in A$ покрывается хотя бы одной траекторией реализации, поскольку справедливо свойство достижимости.

Построим алгебру

$$B = \langle \mathbf{r}, \cup, \& \rangle,$$

где $\mathbf{r} = \{r_{0,0}, \dots, r_{i,j}, \dots, r_{J+1,J+1}, \emptyset\}$ – носитель, $\&$ – операция конкатенации; $\cup$ – операция объединения. Для алгебры справедливы следующие аксиомы:

- аксиома ассоциативности – $r_{i,j} \cup r_{k,l} \cup r_{x,y} = (r_{i,j} \cup r_{k,l}) \cup r_{x,y} = r_{i,j} \cup (r_{k,l} \cup r_{x,y})$;
 $r_{i,j} \& r_{k,l} \& r_{x,y} = (r_{i,j} \& r_{k,l}) \& r_{x,y} = r_{i,j} \& (r_{k,l} \& r_{x,y})$;
- аксиома коммутативности операции объединения – $r_{i,j} \cup r_{k,l} = r_{k,l} \cup r_{i,j}$;
- аксиома дистрибутивности конкатенации относительно объединения –
 $r_{x,y} \& (r_{i,j} \cup r_{k,l}) = (r_{x,y} \& r_{i,j}) \cup (r_{x,y} \& r_{k,l})$;
- аксиома объединения с пустым элементом – $r_{i,j} \cup \emptyset = r_{i,j}$;
- аксиома конкатенации с пустым элементом – $r_{i,j} \& \emptyset = \emptyset \& r_{i,j} = \emptyset$;
- аксиома конкатенации кортежей

$$r = (a_i, a_j) \text{ и } s = (a_j, a_k) \rightarrow r \& s = (a_i, a_j) \& (a_j, a_k) = (a_i, a_j, a_k). \quad (2)$$

Введенные аксиомы позволяют определить конкатенацию $X \times Y$ матрицы смежности $\mathbf{q} = (q_{x,i})$ и $Y \times X$ матрицы смежности $\mathbf{s} = (s_{i,x})$

$$\mathbf{q} \& \mathbf{s} = \left[\bigcup_{i=1}^Y q_{x,i} \& s_{i,x} \right]. \quad (3)$$

Зависимость (3) показывает, что операция конкатенации производится над матрицами смежности по правилам умножения числовых матриц [15], в которых операция умножения « $\times$ » заменяется операцией « $\&$ » конкатенации, а операция суммирования « $+$ » заменяется операцией « $\cup$ » объединения.

С учетом (2) и (3) с помощью операции возведения матрицы r в $(M+1)$ -ю степень конкатенации полумарковская модель (1) может быть преобразована (рис. 1) в

модель, состоящую из линейризованных траекторий реализации, структура которой показана на рис. 2:

$$\mu' = \{A', \mathbf{r}', \mathbf{h}'(t)\}. \quad (4)$$

Множество A' полумарковского процесса включает:

- подмножество операторов/состояний $A'_{J+1} = \{a_0, a_{1(1)}, \dots, a_{i(1)}, \dots, a_{J(1)}, a_{J+1,1}, \dots, a_{1(n)}, \dots, a_{i(n)}, \dots, a_{J(n)}, a_{J+1,n}, \dots, a_{1(N)}, \dots, a_{i(N)}, \dots, a_{J(N)}, a_{J+1(N)}\}$ $a_{i(a)} \in A$, $1 \leq n \leq N$, входящих в завершённые траектории реализации длиной не более чем $M + 1$;
- подмножества операторов/состояний, $A'_i = \{a_0, a_{1[1(i)]}, \dots, a_{m[1(i)]}, \dots, a_{M[1(i)]}, \dots, a_{1[k(i)]}, \dots, a_{m[k(i)]}, \dots, a_{M[k(i)]}, \dots, a_{1[K(i)]}, \dots, a_{m[K(i)]}, \dots, a_{M[K(i)]}, \dots, a_{i[k(i)]}\}$, $a_{m[k(i)]} \in A$, $1 \leq m \leq M$, $1 \leq i \leq J$, входящих в формально незавершённые в операторе «Конец», но заканчивающиеся в операторах $a_1, \dots, a_i, \dots, a_J$ траектории реализации с фиксированной длиной $M + 1$.

В подмножестве A'_{J+1} применена индекс-функция $i(n)$, которая определяет номер i оператора/состояния в n -м пути, ведущем из a_0 в $a_{J+1,n}$. В подмножестве A'_i применена индекс-функция $m[k(i)]$, которая определяет номер m оператора/состояния в k -м пути, ведущем из a_0 в $a_{i,k(i)}$.

Клеточная матрица смежности $\mathbf{r}'$ имеет размер $\Lambda \times \Lambda$, где

$$\Lambda = 1 + \sum_{n=1}^N [J(n) + 1] + (M + 1) \sum_{i=1}^J K(i).$$

Нулевая строка клеточной матрицы смежности $\mathbf{r}'$ имеет вид

$$\begin{aligned} \mathbf{r}'_0 = & [\emptyset, (a_0, a_{1(1)}), \emptyset, \dots, \emptyset, (a_0, a_{1(n)}), \emptyset, \dots, \emptyset, (a_0, a_{1(N)}), \emptyset, \dots, \emptyset, (a_0, a_{1[1(1)]}), \\ & \emptyset, \dots, \emptyset, (a_0, a_{1[k(1)]}), \emptyset, \dots, \emptyset, (a_0, a_{1[K(1)]}), \emptyset, \dots, \emptyset, (a_0, a_{1[1(i)]}), \emptyset, \dots, \emptyset, (a_0, a_{1[k(i)]}), \\ & \emptyset, \dots, \emptyset, (a_0, a_{1[K(i)]}), \emptyset, \dots, \emptyset, (a_0, a_{1[1(J)]}), \emptyset, \dots, \emptyset, (a_0, a_{1[k(J)]}), \emptyset, \dots, \emptyset, (a_0, a_{1[K(J)]}), \\ & \emptyset, \dots, \emptyset], \end{aligned}$$

где нулевой пустой элемент порождает в матрице смежности клетки размером $1 \times *$ по строкам и $* \times 1$ по столбцам; $*$ – произвольный размер; кортежи $[(a_0, a_{1(n)}), \emptyset, \dots, \emptyset]$, $1 \leq n \leq N$ при пересечении с главной диагональю дают N квадратов размером $[J(n) + 1] \times [J(n) + 1]$, порождающих клетки размером $[J(n) + 1] \times *$ по строкам и $* \times [J(n) + 1]$ по столбцам; кортежи $[(a_0, a_{1[k(i)]}), \emptyset, \dots, \emptyset]$, $1 \leq k(i) \leq K(i)$, $1 \leq i \leq J$ при пересечении с главной диагональю дают каждый $K(i)$ квадратов размером $(M + 1) \times (M + 1)$, порождающий клетки $(M + 1) \times *$ по строкам и $* \times (M + 1)$ по столбцам.

Все клетки, кроме тех, которые лежат на главной диагонали, заполняются пустыми элементами, а квадратные матрицы с номерами $1 \leq z \leq Z$, лежащие на главной диагонали, имеют вид

$$\mathbf{r}'_z = \begin{bmatrix} \emptyset & (a_{1(z)}, a_{2(z)}) & \dots & \emptyset & \dots & \emptyset & \emptyset \\ \emptyset & \emptyset & \dots & \emptyset & \dots & \emptyset & \emptyset \\ & & \dots & & & & \\ \emptyset & \emptyset & \dots & (a_{x(z)}, a_{x(z)+1}) & \dots & \emptyset & \emptyset \\ & & \dots & & & & \\ \emptyset & \emptyset & \dots & \emptyset & \dots & \emptyset & (a_{X(z)}, a_{X(z)+1}) \\ \emptyset & \emptyset & \dots & \emptyset & \dots & \emptyset & \emptyset \end{bmatrix}, \quad (5)$$

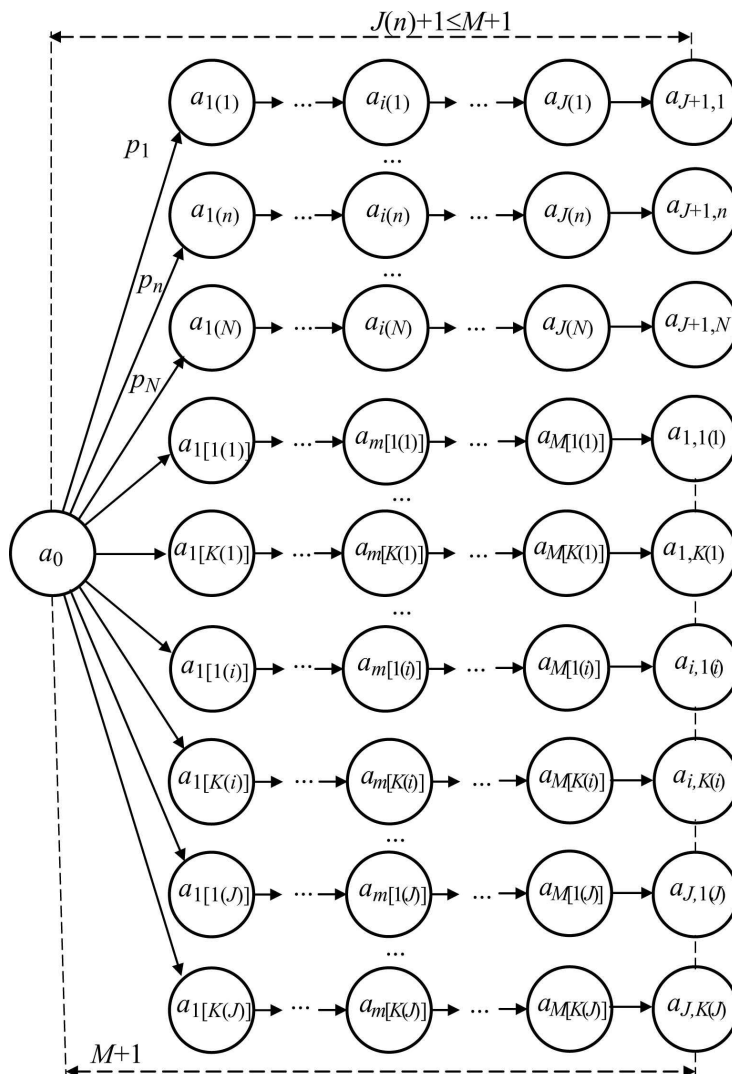


Рис. 2. Структура алгоритма, состоящего из линейризованных траекторий

где $1 \leq z \leq N$ – номера квадратов, описывающих нормально завершённые траектории реализации длиной не более чем $M + 1$; $z(i, k)$ – номера квадратов, описывающих формально незавершённые траектории реализации длиной $M + 1$, заканчивающиеся в операторах/состояниях $a_1, \dots, a_i, \dots, a_J$; $a_{x(z)} \in \{a_1, \dots, a_i, \dots, a_J\}$ – состояния, включаемые в пути в соответствии с (3) при возведении матрицы $\mathbf{r} = [r_{i,j}]$ в степень конкатенации (рис. 1);

$$\left[2 + N + k(i) + \sum_{j=1}^{i-1} K(j) \leq z(i, k) \leq 2 + N + \sum_{j=1}^i K(j) \right]. \quad (6)$$

Полумарковская матрица $\mathbf{h}'(t)$ получается из $\mathbf{h}(t)$ через применение к ней прямого преобразования Лапласа, возведения получившейся характеристической матрицы в степень, выделение из матрицы необходимых элементов и применение к ним обратного преобразования Лапласа [16–18]. Но сформированная с помощью операций (2), (3) матрица смежности включает в основном пустые элементы, поэтому более продуктивной является методика, основанная на замене пустых элементов матрицы $\mathbf{r}'$

алгебраическими нулями, а непустых элементов первой строки и квадратов, стоящих на главной диагонали $\mathbf{h}'(t)$, – элементами, формируемыми из элементов матрицы плотностей распределения $\mathbf{f}(t)$ и стохастической матрицы p (рис. 2).

Согласно методике, нулевая строка полумарковской матрицы $\mathbf{h}'(t)$ имеет вид

$$\mathbf{h}'_0(t) = [0, p_{0,1(1)}\delta(t), 0, \dots, 0, p_{0,1(n)}\delta(t), 0, \dots, 0, p_{0,1(N)}\delta(t), 0, \dots, 0, p_{0,1[1(1)]}\delta(t), 0, \dots, 0, p_{0,1[k(1)]}\delta(t), 0, \dots, 0, p_{0,1[K(1)]}\delta(t), 0, \dots, 0, p_{0,1[1(i)]}\delta(t), 0, \dots, 0, p_{0,1[k(i)]}\delta(t), 0, \dots, 0, p_{0,1[K(i)]}\delta(t), 0, \dots, 0, p_{0,1[1(J)]}\delta(t), 0, \dots, 0, p_{0,1[k(J)]}\delta(t), 0, \dots, 0, p_{0,1[K(J)]}\delta(t), 0, \dots, 0], \quad (7)$$

где $\delta(t)$ – δ -функция Дирака; $p_{0,1(n)} \in \mathbf{p}$ – вероятность того, что n -я завершенная траектория реализации начинается в операторе/состоянии $a_{1(n)} \in A$ и выстраивается в соответствии с порядком, определенным матричной конкатенацией (2) для конкретной траектории. Ненулевые клетки полумарковской матрицы имеют вид

$$\mathbf{h}'_z(t) = \begin{bmatrix} 0 & f_{1(z)}(t) & \dots & 0 & \dots & 0 & 0 \\ 0 & 0 & \dots & 0 & \dots & 0 & 0 \\ & & & \dots & & & \\ 0 & 0 & \dots & f_{x(z)}(t) & \dots & 0 & 0 \\ & & & \dots & & & \\ 0 & 0 & \dots & 0 & \dots & 0 & f_{X(z)}(t) \\ 0 & 0 & \dots & 0 & \dots & 0 & 0 \end{bmatrix},$$

где $f_{x(z)}(t)$ – плотность распределения времени пребывания алгоритма/процесса в $x(z)$ -м состоянии z -й линейной траектории реализации.

Плотности распределения $f_{x(z)}(t)$, входящие в (7), и вероятности, формирующие веса δ -функций Дирака в (6), могут быть получены из $\mathbf{f}(t)$ следующим образом. Выберем в матрице (5) непустой элемент, например $(a_{x(z)}, a_{x(z)+1})$. Пусть в выбранном непустом элементе $a_{x(z)}$ соответствует $a_i \in A$, а $a_{x(z)+1}$ соответствует $a_j \in A$. Тогда $h_{x(z)}(t) p_{x(z)} = p_{i,j}$, $f_{x(z)}(t) = f_{i,j}(t)$ (рис. 1). Отсюда следует, что вероятность выбора z -й траектории реализации алгоритма/полумарковского процесса и плотность распределения времени переключений по операторам/состояниям траектории равны соответственно

$$p_z = \prod_{x(z)=1}^{X(z)+1} p_{x(z)}, f_z(t) = L^{-1} \left[\prod_{x(z)=1}^{X(z)+1} L [f_{x(z)}(t)]^{x(z)} \right], \quad (8)$$

где $X(z) = J(n) + 1$ – для завершенных траекторий реализации, если $1 \leq z = n \leq N$; $X(z) = M + 1$ – для незавершенных траекторий реализации, если $z(i, k)$ определяется по зависимости (6); $p_z \in P'$; P' – множество вероятностей появления соответствующих траекторий реализации в полумарковской модели (4); $L[\dots]$ и $L^{-1}[\dots]$ – прямое и обратное преобразование Лапласа соответственно;

$$P' = \{p_{0,1(1)}, \dots, p_{0,1(n)}, \dots, p_{0,1(N)}, p_{0,1[1(1)]}, \dots, p_{0,1[k(1)]}, \dots, p_{0,1[K(1)]}, \dots, p_{0,1[1(i)]}, \dots, p_{0,1[k(i)]}, \dots, p_{0,1[K(i)]}, \dots, p_{0,1[1(J)]}, \dots, p_{0,1[k(J)]}, \dots, p_{0,1[K(J)]}\}. \quad (9)$$

Вследствие того, что вероятности $p_{i,j} \leq \mathbf{p}$ формируются по логическим условиям, определенным в операторах принятия решения алгоритма обработки данных, и по распределениям обрабатываемых случайных данных, появление траекторий реализации составляет полную группу несовместных событий. Поэтому

$$\sum_{p_z \in P'} p_z = 1.$$

Вероятности (9) и плотности распределения времени реализации траекторий (8) являются ключевыми характеристиками надежности ПО.

2. Трансформация линеаризованной модели в модель отказов программного обеспечения

Полученная линеаризованная модель алгоритма/полумарковского процесса позволяет выявлять ошибки, допущенные программистом при разработке ПО при формировании структуры и параметров алгоритма обработки данных [19–21].

К ошибкам при формировании структуры относятся:

- наличие операторов или групп операторов, порождающих автономные и полу-автономные первого и второго рода циклы;
- нарушение порядка следования операторов в текущей траектории реализации;
- пропуск операторов в текущей траектории реализации;
- появление лишних операторов в текущей траектории реализации.

К параметрическим ошибкам относятся:

- выход заданного времени между интерпретациями двух операторов при реализации алгоритма (перекося данных, чистое запаздывание) за установленные в техническом задании пределы;
- выход точности обработки поступающих данных за установленные в техническом задании пределы;
- неверное определение формата обрабатываемых данных;
- неверное определение условий выбора ветвей алгоритма в местах ветвления.

Перечисленные ошибки выявляются при анализе конкретных линейных траекторий реализации алгоритма/полумарковского процесса (4). Поэтому вероятности появления указанных ошибок в реализациях алгоритма совпадают с вероятностями появления соответствующих линейных ветвей алгоритма. Разделим множество на три непересекающихся подмножества:

$$\begin{aligned} P' &= P'_{\text{true}} \cup P'_{\text{error_s}} \cup P'_{\text{error_p}}, P'_{\text{true}} \cap P'_{\text{error_s}} = \emptyset; \\ P'_{\text{error_s}} \cap P'_{\text{error_p}} &= \emptyset; \\ P'_{\text{true}} \cup P'_{\text{error_p}} &= \emptyset, \end{aligned} \quad (10)$$

где P'_{true} – множество вероятностей появления правильной реализации алгоритма/полумарковского процесса; $P'_{\text{error_s}}$ – множество вероятностей появления реализации со структурными и параметрическими ошибками; $P'_{\text{error_p}}$ – множество вероятностей появления реализации с параметрическими ошибками.

Будем считать, что структурные ошибки всегда приводят к отказу ПО, а параметрические ошибки могут приводить, а могут и не приводить к отказу. Например, если программа содержит итерационный цикл, то возможно разделение области обрабатываемых данных, определяющих структуру этого цикла, на две подобласти, в одной из которых заданная точность обеспечивается, а в другой – нет. Это, в свою очередь, означает, что

$$P'_{\text{error_p}} = P'_{\text{error_p},1} \cup P'_{\text{error_p},2}, P'_{\text{error_p},1} \cap P'_{\text{error_p},2} = \emptyset,$$

где $P'_{\text{error_p},1}$ – вероятность появления ошибки, не приводящей к отказу; $P'_{\text{error_p},2}$ – вероятность появления ошибки, приводящей к отказу.

Вероятности появления безотказной и отказной реализаций алгоритма/полумарковского процесса определяются следующим образом:

$$p = \sum_{p_z \in (P'_{\text{true}} \cup P'_{\text{error_p},1})} p_z; \quad q = \sum_{p_z \in (P'_{\text{error_s}} \cup P'_{\text{error_p},2})} p_z; \quad p + q = 1, \quad (11)$$

где p – вероятность появления безотказной реализации; q – вероятность появления отказной реализации.

Полумарковская модель надежности ПО имеет вид (рис. 3)

$$\mu_{\text{failure}} = [B, \mathbf{r}_{\text{failure}}, \mathbf{h}_{\text{failure}}(t)], \quad (12)$$

где $B = \{\beta_0, \beta_1, \beta_2, \beta_3\}$ – множество состояний; β_0 – состояние, моделирующее запуск программы, β_1 – состояние, моделирующее исполнение программы и идентификацию безотказной и отказной реализаций; β_2 – состояние, фиксирующее безотказную реализацию; β_3 – состояние, фиксирующее отказную реализацию; $\mathbf{r}_{\text{failure}}$ – матрица смежности; $\mathbf{h}_{\text{failure}}(t)$ – полумарковская матрица;

$$\mathbf{r}_{\text{failure}} = \begin{bmatrix} \emptyset & (\beta_0, \beta_1) & \emptyset & \emptyset \\ \emptyset & \emptyset & (\beta_1, \beta_2) & (\beta_1, \beta_3) \\ (\beta_2, \beta_0) & \emptyset & \emptyset & \emptyset \\ (\beta_3, \beta_0) & \emptyset & \emptyset & \emptyset \end{bmatrix};$$

$$\mathbf{h}_{\text{failure}}(t) = \begin{bmatrix} \emptyset & \delta(t) & \emptyset & \emptyset \\ \emptyset & \emptyset & h_{p_z \in (P'_{\text{true}} \cup P'_{\text{error_p},1})}(t) & h_{p_z \in (P'_{\text{error_s}} \cup P'_{\text{error_p},2})}(t) \\ \delta(t) & \emptyset & \emptyset & \emptyset \\ \delta(t) & \emptyset & \emptyset & \emptyset \end{bmatrix};$$

$$h_{p_z \in (P'_{\text{true}} \cup P'_{\text{error_p},1})}(t) = \sum_{p_z \in (P'_{\text{true}} \cup P'_{\text{error_p},1})} p_z f_z(t);$$

$$h_{p_z \in (P'_{\text{error_s}} \cup P'_{\text{error_p},2})}(t) = \sum_{p_z \in (P'_{\text{error_s}} \cup P'_{\text{error_p},2})} p_z f_z(t).$$

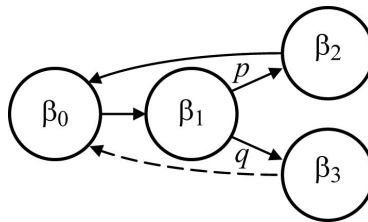


Рис. 3. Полумарковская модель отказов программного обеспечения

Из модели (12) следует, что распределение вероятностей по количеству отказных реализаций y среди общего количества реализаций Y определяется биномиальным законом распределения [21–23]

$$P_{y,Y} = C_Y^y p^{Y-y} q^y, \quad (13)$$

где $C_Y^y = \frac{Y!}{y!(Y-y)!}$; q и p – вероятность отказа и безотказной работы программы при единичном запуске соответственно.

Вид распределения (13) для случая $Y = 100$ запусков $q = 0,0025, 0,005, 0,01, 0,02, 0,04, 0,08$ показан на рис. 4.

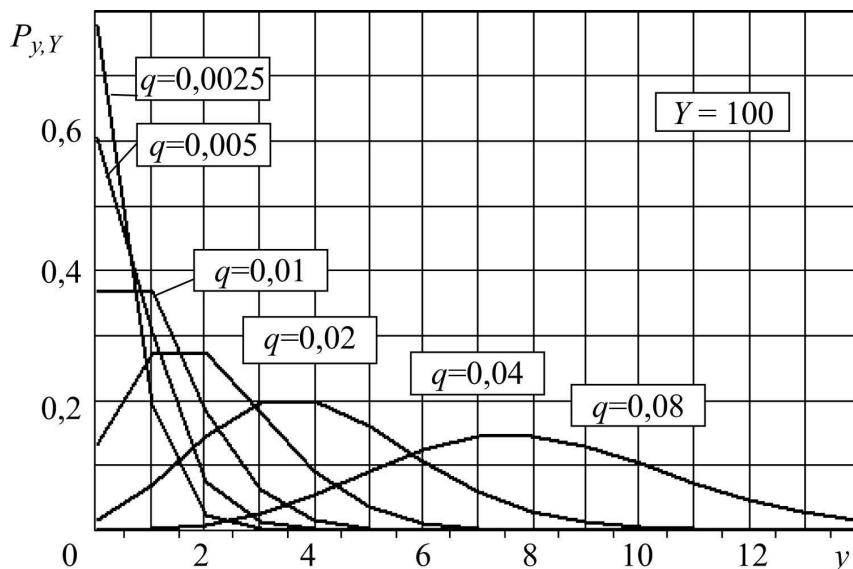


Рис. 4. Распределение вероятностей появления y отказов программного обеспечения при $Y = 100$ запусках программы

Следует отметить, что в (13) при расчете оценки надежности ПО аргумент является дискретным и физически означает количество запусков программ [24]. Кроме того, вид полученных кривых существенно зависит от количества проведенных экспериментов Y . В то же время в теории надежности все смены состояний анализируемых систем привязаны к физическому времени, причем единственным параметром, по которому оценивается надежность, является плотность потока отказов [25, 26]. Поэтому устремим $Y \rightarrow \infty$, $q \rightarrow 0$ таким образом, что $Yq = \alpha = \text{const}$. В этом случае [27]

$$\lim_{Y \rightarrow \infty} P_{y,Y} = C_Y^y p^{Y-y} q^y = P_\alpha = \frac{\alpha^y}{y!} \exp(-\alpha),$$

где α – параметр закона Пуассона.

Единственным времязадающим фактором в полумарковской модели (12) являются взвешенная плотность распределения $h_{p_z \in (P'_{\text{true}} \cup P'_{\text{error_p},1})}(t)$ времени безотказной штатной реализации алгоритма/полумарковского процесса и взвешенная плотность распределения $h_{p_z \in (P'_{\text{error_s}} \cup P'_{\text{error_p},2})}(t)$ отказной его реализации. С учетом этого время наработки ПО на отказ может быть определено как среднее время достижения состояния β_3 из состояния β_0 при отключенной обратной связи β_3, β_0 , показанной на рис. 3 пунктирной стрелкой:

$$T_{\text{STBF}} = \frac{T_{p_z \in (P'_{\text{true}} \cup P'_{\text{error_p},1})} \cdot p}{q} + T_{p_z \in (P'_{\text{error_s}} \cup P'_{\text{error_p},2})}, \quad (14)$$

где T_{STBF} – время наработки ПО на отказ (Software Time Before Failure); $T_{p_z \in (P'_{\text{true}} \cup P'_{\text{error_p},1})}$ – математическое ожидание времени безотказных реализаций алгоритма/полумарковского процесса; $T_{p_z \in (P'_{\text{error_s}} \cup P'_{\text{error_p},2})}$ – математическое ожидание

времени отказных реализаций алгоритма/полумарковского процесса;

$$T_{p_z \in (P'_{\text{true}} \cup P'_{\text{error_p},1})} = \frac{\int_0^\infty t \cdot h_{p_z \in (P'_{\text{true}} \cup P'_{\text{error_p},1})}(t)}{p};$$

$$T_{p_z \in (P'_{\text{error_s}} \cup P'_{\text{error_p},2})} = \frac{\int_0^\infty t \cdot h_{p_z \in (P'_{\text{error_s}} \cup P'_{\text{error_p},2})}(t)}{q}.$$

Поток безотказных реализаций алгоритма/полумарковского процесса представляет собой сумму потоков с разными значениями времени реализации, следовательно, в соответствии с теоремой Б.И. Григелиониса [28] он стремится к пуассоновскому. Плотность распределения времени между двумя отказами ПО в пуассоновском потоке определяется экспоненциальным законом распределения

$$f_{\text{STBF}}(t) \cong \frac{1}{T_{\text{STBF}}} \exp\left(-\frac{t}{T_{\text{STBF}}}\right),$$

где T_{STBF} – рассчитывается по зависимости (14).

В соответствии с (14) время наработки ПО до отказа напрямую связывается с вероятностями p и q . Если режим эксплуатации программного обеспечения таков, что при переключении (β_3, β_3) производится исправление ошибок, то в соответствии с (10) коррекция алгоритмического и программного обеспечения приводит к сокращению области $P'_{\text{error_s}} \cup P'_{\text{error_p},2}$ и расширению области $P'_{\text{true}} \cup P'_{\text{error_p},1}$. Это, в соответствии с (11) приводит к уменьшению q и увеличению p , а, следовательно, по мере выявления и устранения ошибок в программном обеспечении поток виртуальных отказов в системе сокращается.

Заключение

Основными результатами исследования следует считать метод линеаризации алгоритмов обработки данных в аппаратно-программных комплексах и метод трансформации линеаризованной модели в модель отказов ПО с расчетом параметров модели отказов по структурным особенностям алгоритма и параметрам обрабатываемых данных. Дальнейшее продолжение работ в этом направлении связывается с развитием методов выявления структурных и параметрических особенностей в линеаризованных моделях алгоритмов, а также связи статистики обрабатываемых случайных данных со статистикой генерации отказов.

Исследование выполнено при поддержке гранта Российского научного фонда (проект № 25-29-20177).

Литература

1. Dubrova, E. Fault Tolerant Design / E. Dubrova. – Switzerland: Springer, 2013.
2. Богомолов, А.В. Математические модели программных отказов цифровых систем управления / А.В. Богомолов, Е.В. Ларкин, А.Н. Привалов // Научно-техническая информация. Серия 2: Информационные процессы и системы. – 2025. – № 1. – С. 1–9.
3. Larkin, E. Digital Control of Continuous Production with Dry Friction at Actuators / E. Larkin, A. Privalov, A. Bogomolov, T. Akimenko // Smart Innovation, Systems and Technologies. – 2022. – V. 232. – P. 427–436.
4. Привалов, А.Н. Моделирование надежности программных компонентов киберфизических систем / А.Н. Привалов, Е.В. Ларкин, А.В. Богомолов // Программные продукты и системы. – 2025. – № 1. – С. 47–54.

5. Aital, P. Role of Software Reliability Models in Performance Improvement and Management / P. Aital, P. Sashikala // Journal of Software Engineering and Applications. – 2021. – V. 5, № 9. – P. 737–742.
6. Sanchez-Silva, M. Reliability and Life-Cycle Analysis of Deteriorating Systems / M. Sanchez-Silva, G.-A. Klutke. – Switzerland: Springer, 2016.
7. Naes, A. System Reliability Analysis by Enhanced Monte Carlo Simulation / A. Naes, B.J. Leira, O. Batsevych // Structural Safety. – 2009. – V. 31, № 5. – P. 349–355.
8. Golosovskiy, M.S. Algorithm for Setting Fuzzy Logical Inclusion Systems Based on Statistical Data / M.S. Golosovskiy, A.V. Bogomolov, D.S. Tobin // Automatic Documentation and Mathematical Linguistics. – 2023. – V. 57, № 1. – P. 1–9.
9. Larkin, E. Reliability of Robot's Controller Software / E. Larkin, T. Akimenko, A. Bogomolov, V. Sharov // Interactive Collaborative Robotics. Lecture Notes in Computer Science. – 2023. – V. 14214. – P. 289–299.
10. Shatnawi, O. Measuring Commercial Software Operational Reliability. An Interdisciplinary Modelling Approach / O. Shatnawi // Eksploatacja i Niezawodnose – Maintenance and Reliability. – 2014. – V. 16, № 4. – P. 585–594.
11. Shatnawi, O. Testing-Effort Dependent Software Reliability Model for Distributed Systems / O. Shatnawi // International Journal of Distributed Systems and Technologies. – 2013. – № 4. – P. 1–14.
12. Janssen, J. Applied Semi-Markov Processes / J. Janssen, R. Manca. – New York: Springer, 2006.
13. Jiang, Q. Event-Driven Semi-Markov Switching State-Space Control Processes / Q. Jiang, H.-S. Xi, B.-Q. Yin // IET Control Theory and Applications. – 2012. – V. 6, № 12. – P. 1861–1869.
14. Bielecki, T.R. Conditional Markov Chains: Properties, Construction and Structured Dependence / T.R. Bielecki, J. Jakubowski, M. Nieweglowski // Stochastic Processes and Their Applications. – 2017. – V. 127, № 4. – P. 1125–1170.
15. Cherney, D. Linear Algebra / D. Cherney, T. Denton, R. Thomas, A. Waldron. – California: Davis, 2013.
16. Pavlov, A.V. About the Equality of the Transform of Laplace to the Transform of Fourier / A.V. Pavlov // Issues of Analysis. – 2016. – V. 5 (23), № 4 (76). – P. 21–30.
17. Jianhui Li. Three Effective Inverse Laplace Transform Algorithms for Computing Time-Domain Electromagnetic Responses / Jianhui Li, C.G. Farquharson, Hu Xiangyun // Geophysics. – 2015. – V. 81, № 2. – P. 75–90.
18. Larkin, E.V. Discrete Approach to Simulating Synchronized Relay Races / E.V. Larkin, A.N. Privalov, A.V. Bogomolov // Automatic Documentation and Mathematical Linguistics. – 2020. – V. 54, № 1. – P. 43–51.
19. Barack, O. Assessment and Prediction of Software Reliability in Mobile Applications / O. Barack, L. Huang // Journal of Software Engineering and Applications. – 2020. – V. 13, № 9. – P. 179–190.
20. Inoue, S. Lognormal Process Software Reliability Modeling with Testing Effort / S. Inoue, S. Yamada // Journal of Software Engineering and Applications. – 2013. – V. 6, № 4. – P. 8–14.
21. Kobayashi, H. Probability, Random Processes and Statistical Analysis / H. Kobayashi, B.L. Marl, W. Turin. – London: Cambridge University Press, 2012.
22. Tobin, D. Model of Organization of Software Testing for Cyber-Physical Systems / D. Tobin, A. Bogomolov, M. Golosovskiy // Cyber-Physical Systems: Modelling and Industrial Application. Studies in Systems, Decision and Control. – Switzerland: Springer, 2022. – V. 418. – P. 51–60.

23. Ларкин, Е.В. Эстафеты по выбранным альтернативным маршрутам / Е.В. Ларкин, А.В. Богомолов, А.Н. Привалов, Н.Н. Добровольский // Вестник ЮУрГУ. Серия: Математическое моделирование и программирование. – 2018. – Т. 11, № 1. – С. 15–26.
24. Ларкин, Е.В. Дискретная модель парных эстафет / Е.В. Ларкин, А.В. Богомолов, А.Н. Привалов, Н.Н. Добровольский // Вестник ЮУрГУ. Серия: Математическое моделирование и программирование. – 2018. – Т. 11, № 3. – С. 72–84.
25. Larkin, E. Discrete Model of Mobile Robot Assemble Fault-Tolerance / E. Larkin, A. Bogomolov, A. Privalov // Interactive Collaborative Robotics. Lecture Notes in Computer Science. – 2019. – V. 11659. – P. 204–215.
26. Ларкин, Е.В. Моделирование надежности бортового оборудования мобильного робота / Е.В. Ларкин, Т.А. Акименко, А.В. Богомолов // Известия Саратовского университета. Новая серия. Серия: Математика. Механика. Информатика. – 2021. – Т. 21, № 3. – С. 390–399.
27. Hongyuan Lu. A Functional Central Limit Theorem for Markov Additive Arrival Processes and its Applications to Queueing Systems / Hongyuan Lu, Guodong Pang, M. Mandjes // Queueing Systems. – 2016. – V. 84, № 3. – P. 381–406.
28. Григелионис, Б.И. О сходимости сумм ступенчатых случайных процессов к пуассоновскому / Б.И. Григелионис // Теория вероятностей и ее применения. – 1963. – Т. 8, № 2. – С. 189–194.

Евгений Васильевич Ларкин, доктор технических наук, профессор, профессор-консультант, Тульский государственный университет (г. Тула, Российская Федерация), elarkin@mail.ru.

Алексей Валерьевич Богомолов, доктор технических наук, профессор, ведущий научный сотрудник, Федеральный исследовательский центр «Информатика и управление» Российской академии наук (г. Москва, Российская Федерация), a.v.bogomolov@gmail.com.

Александр Николаевич Привалов, доктор технических наук, профессор, директор института передовых информационных технологий, Тульский государственный педагогический университет имени Л.Н. Толстого (г. Тула, Российская Федерация), privalov.61@mail.ru.

Поступила в редакцию 9 апреля 2025 г.

MSC 68U20

DOI: 10.14529/mmp250307

MATHEMATICAL MODELLING OF SOFTWARE FAILURES

E. V. Larkin¹, A. V. Bogomolov², A. N. Privalov³

¹Tula State University, Tula, Russian Federation

²Federal Research Center “Computer Science and Control” of the Russian Academy of Sciences, Moscow, Russian Federation

³Tolstoy Tula State Pedagogical University, Tula, Russian Federation

E-mail: elarkin@mail.ru, a.v.bogomolov@gmail.com, privalov.61@mail.ru

The problematic issues of modeling evaporation-condensation in the numerical study of non-stationary heat and mass transfer in cryogenic processes are considered. The statement

of the problem of calculating the parameters of a multi-phase turbulent flow in a closed region is given when approximating the free surface by the liquid volume method, the use of which makes it possible to track the vertical displacement of the phase boundary in the process of changing the cryoprotectant parameters over time. A description of evaporation-condensation models suitable for use in non-stationary calculations of flows in a closed region together with the liquid volume method is given. The possibility of using the Lee evaporation-condensation model and the model based on the law of particle diffusion is demonstrated as applied to calculations on simple computational grids. The results of calculations of the change in the pressure of a cryogenic product during non-drainage storage using various models of boiling-condensation are presented. The obtained simulation results were used in the accumulation of a database of storage parameters in the system for remote monitoring of the state of cryogenic equipment. The data are estimated storage times for various reference pressures and liquid levels in the vessel, with each external heat flux value having a specific estimated no-drain storage time.

Keywords: heat and mass transfer; non-drainage storage; computational fluid dynamics; evaporation-condensation model; tank container.

References

1. Dubrova E. *Fault Tolerant Design*. Switzerland, Springer, 2013.
2. Bogomolov A.V., Larkin E.V., Privalov A.N. Mathematical Models of Software Failures of Digital Control Systems. *Scientific and Technical Information. Series 2: Information Processes and Systems*, 2025, no. 1, pp. 1–9.
3. Larkin E., Privalov A., Bogomolov A., Akimenko T. Digital Control of Continuous Production with Dry Friction at Actuators. *Smart Innovation, Systems and Technologies*, 2022, vol. 232, pp. 427–436.
4. Privalov A.N., Larkin E.V., Bogomolov A.V. Reliability Modeling of Software Components of Cyber-Physical Systems. *Software Products and Systems*, 2025, no. 1, pp. 47–54.
5. Aital P., Sashikala P. Role of Software Reliability Models in Performance Improvement and Management. *Journal of Software Engineering and Applications*, 2021, vol. 5, no. 9, pp. 737–742.
6. Sanchez-Silva M., Klutke G.-A. *Reliability and Life-Cycle Analysis of Deteriorating Systems*. Switzerland, Springer, 2016.
7. Naes A., Leira B.J., Batsevych O. System Reliability Analysis by Enhanced Monte Carlo Simulation. *Structural Safety*, 2009, vol. 31, no. 5, pp. 349–355.
8. Golosovskiy M.S., Bogomolov A.V., Tobin D.S. Algorithm for Setting Fuzzy Logical Inclusion Systems Based on Statistical Data. *Automatic Documentation and Mathematical Linguistics*, 2023, vol. 57, no 1, pp. 1–9.
9. Larkin E., Akimenko T., Bogomolov A., Sharov V. Reliability of Robot's Controller Software. *Interactive Collaborative Robotics. Lecture Notes in Computer Science*, 2023, vol. 14214, pp. 289–299.
10. Shatnawi O. Measuring Commercial Software Operational Reliability. An Interdisciplinary Modelling Approach. *Eksplatacia i Niezawodnosc – Maintenance and Reliability*, 2014, vol. 16, no. 4, pp. 585–594.
11. Shatnawi O. Testing-Effort Dependent Software Reliability Model for Distributed Systems. *International Journal of Distributed Systems and Technologies*, 2013, no. 4, pp. 1–14.
12. Janssen J., Manca R. *Applied Semi-Markov Processes*. New York, Springer, 2006.
13. Q. Jiang, H.-S. Xi, B.-Q. Yin Event-Driven Semi-Markov Switching State-Space Control Processes. *IET Control Theory and Applications*, 2012, vol. 6, no. 12, pp. 1861–1869. DOI: 10.1049/iet-cta.2011.0174
14. Bielecki T.R., Jakubowski J., Nieweglowski M. Conditional Markov Chains: Properties, Construction and Structured Dependence. *Stochastic Processes and Their Applications*, 2017, vol. 127, no. 4, pp. 1125–1170.

15. Cherney D., Denton T., Thomas R. *Linear Algebra*. California, Davis, 2013.
16. Pavlov A.V. About the Equality of the Transform of Laplace to the Transform of Fourier. *Issues of Analysis*, 2016, vol. 5 (23), no. 4 (76), pp. 21–30.
17. Jianhui Li, C.G. Farquharson, Hu Xiangyun. Three Effective Inverse Laplace Transform Algorithms for Computing Time-Domain Electromagnetic Responses. *Geophysics*, 2015, vol. 81, no. 2, pp. 75–90.
18. Larkin E.V., Privalov A.N., Bogomolov A.V. Discrete Approach to Simulating Synchronized Relay Races. *Automatic Documentation and Mathematical Linguistics*, 2020, vol. 54, no. 1, pp. 43–51.
19. Barack O., Huang L. Assessment and Prediction of Software Reliability in Mobile Applications. *Journal of Software Engineering and Applications*, 2020, vol. 13, no. 9, pp. 179–190.
20. Inoue S., Yamada S. Lognormal Process Software Reliability Modeling with Testing Effort. *Journal of Software Engineering and Applications*, 2013, vol. 6, no. 4, pp. 8–14.
21. Kobayashi H., Marl B.L., Turin W. *Probability, Random Processes and Statistical Analysis*. London, Cambridge University Press, 2012.
22. Tobin D., Bogomolov A., Golosovskiy M. Model of Organization of Software Testing for Cyber-Physical Systems. *Cyber-Physical Systems: Modelling and Industrial Application. Studies in Systems, Decision and Control*, 2022, vol. 418, pp. 51–60.
23. Larkin E.V., Bogomolov A.V., Privalov A.N., Dobrovolsky N.N. Relay Races Along a Pair of Selectable Routes. *Bulletin of the South Ural State University. Series: Mathematical Modelling, Programming and Computer Software*, 2018, vol. 11, no. 1, pp. 15–26. DOI: 10.14529/mmp180102
24. Larkin E.V., Bogomolov A.V., Privalov A.N., Dobrovolsky N.N. Discrete Model of Paired Relay-Race. *Bulletin of the South Ural State University. Series: Mathematical Modelling, Programming and Computer Software*, 2018, vol. 11, no. 3, pp. 72–84. DOI: 10.14529/mmp180306
25. Larkin E., Bogomolov A., Privalov A. Discrete Model of Mobile Robot Assemble Fault-Tolerance. *Interactive Collaborative Robotics. Lecture Notes in Computer Science*, 2019, vol. 11659, pp. 204–215.
26. Larkin E.V., Akimenko T.A., Bogomolov A.V. Modeling the Reliability of the Onboard Equipment of a Mobile Robot. *Izvestiya of Saratov University. Mathematics. Mechanics. Informatics*, 2021, vol. 21, no. 3, pp. 390–399. DOI: 10.18500/1816-9791-2021-21-3-390-399
27. Lu Hongyuan, Guodong Pang, M. Mandjes A Functional Central Limit Theorem for Markov Additive Arrival Processes and its Applications to Queueing Systems. *Queueing Systems*, 2016, vol. 84, no. 3, pp. 381–406.
28. Grigelionis B.I. On the Convergence of Sums of Step Random Processes to the Poisson Process. *Probability Theory and Its Applications*, 1963, vol. 8, no. 2, pp. 189–194. (in Russian)

Received April 9, 2025