УДК 681.5.09

ЛОГИЧЕСКИЕ МЕТОДЫ В ТЕОРИИ НАДЕЖНОСТИ. II. МАТЕМАТИЧЕСКИЕ МОДЕЛИ НАДЕЖНОСТИ

В.И. Левин

ГОУ ВПО «Пензенская государственная технологическая академия»; levin@pgta.ac.ru

Представлена членом редколлегии профессором В.И. Коноваловым

Ключевые слова и фразы: двоичный оператор; динамический автомат; логическая теория надежности; надежностный процесс; переключательный процесс; структура оператора.

Аннотация: Предложена автоматно-логическая модель надежности систем, где входные процессы автомата моделируют надежностные процессы в блоках системы, а выходные процессы автомата – надежностные процессы в самой системе.

В первой части статьи [1] описаны дискретная двузначная и непрерывная логики, служащие математическим аппаратом для моделирования соответственно статики и динамики надежности.

2.1. Переключательные процессы

Рассмотрим произвольную двоичную функцию непрерывного времени t, генерируемую в некоторой системе (в частности технической системе), то есть функцию $x = x(t)$, значения которой принадлежат множеству $\{0, 1\}$. Пусть эта функция удовлетворяет трем условиям: 1) значение функции x в момент ее изменения a по определению совпадает со значением x при $t > a$; 2) значения x определены на интервале времени $(-\infty, \infty)$; 3) на любом конечном подынтервале указанного интервала имеется конечное число изменений значения функции. Введенная функция называется *переключательным процессом* в системе.

Обозначим: 1 – постоянный процесс, равный единице на некотором интервале времени; 0 – постоянный процесс, равный нулю на некотором интервале времени; $1'$ – изменение значения процесса $0 \to 1$; $0'$ – изменение значения процесса $1 \to 0$; $0'_a$ – изменение $0'$ в момент a; $1'_a$ – изменение $1'$ в момент a; $1(a,b)$ – импульс $1'_a 0'_b$; $0(a,b)$ – пауза $0'_a 1'_b$. По условию 1) в некоторой окрестности момента a изменения значения процесса

$$1'_a = \begin{cases} 0, & t < a; \\ 1, & t \ge a; \end{cases} \quad 0'_a = \begin{cases} 1, & t < a; \\ 0, & t \ge a. \end{cases} \tag{2.1}$$

Согласно (2.1) *импульс* – это интервал единичных значений процесса, включающий начало и не включающий конец, а *пауза* – интервал нулевых значений процесса с аналогичными включениями:

$$1(a,b)=\begin{cases}1, & a\le t<b;\\ 0, & t<a \text{ или } t\ge b;\end{cases} \quad 0(a,b)=\begin{cases}0, & a\le t<b;\\ 1, & t<a \text{ или } t\ge b.\end{cases} \tag{2.2}$$

Формулы (2.2) при $a=b$ принимают вид:

$$1(a,a)\equiv 0; \quad 0(a,a)\equiv 1\,. \tag{2.3}$$

Видим, что импульс (пауза) с совмещенными началом и концом фактически есть отсутствие импульса (паузы), то есть вырожденный участок, который может быть исключен из рассмотрения. Однако из (2.3) следует возможность формально рассматривать отсутствие импульса (паузы), то есть тождественный 0 (тождественную 1), как импульс (паузу) с совмещенными началом и концом, что часто бывает полезно. Отметим также возможность рассматривать изменения процесса (2.1) как импульс (паузу) на бесконечном интервале:

$$1'_a=1(a,\infty)=0(-\infty,a); \quad 0'_a=0(a,\infty)=1(-\infty,a)\,. \tag{2.4}$$

Введем необходимые определения. Пусть $x(t)$ – любой переключательный процесс, отличный от тождественного 0 или 1; a_x – момент первого изменения (начала) и b_x – момент последнего изменения (окончания) $x(t)$, причем оба момента конечны. Значение x_0 процесса при $t<a_x$ назовем его *начальным значением*. При этом будем говорить, что $x(t)$ *начинается* импульсом (паузой), если $x_0=0$ $(x_0=1)$. Аналогично значение x_∞ процесса при $t>b_x$ назовем его *конечным значением*, говоря, что $x(t)$ *оканчивается* импульсом (паузой), если $x_\infty=0$ $(x_\infty=1)$. Процессы $x(t)$, $y(t)$ назовем *непересекающимися во времени*, если $b_x\le a_y$.

Общее число изменений значения переключательного процесса называется *длиной L процесса.* При $L\le 1$ процесс считается *простым*, при $L\ge 2$ – *сложным.* Два переключательных процесса *равны*, если у них одинаковое число соответственно однотипных изменений, моменты которых совпадают. Два переключательных процесса с буквенными моментами изменений считаем *эквивалентными*, если при любой численной конкретизации указанных моментов оба процесса становятся равными.

Будем записывать переключательные процессы в виде последовательности изменений с указанием момента изменения или в виде последовательности импульсов и пауз. Во втором случае для простоты опускаем начальное и конечное постоянные значения, а моменты промежуточных изменений указываем один раз либо в импульсе, либо в соседней паузе. Например, один и тот же процесс можно записать так: $x(t)=1'_a0'_b1'_c0'_d1'_e$ или $x(t)=1(a,b)0(-,c)1(-,d)0(-,e)$.

Этот процесс до момента a равен 0, в интервале $a\le t<b$ он равен 1, в интервале $b\le t<c$ равен 0, в интервале $c\le t<d$ – снова 1, в интервале $d\le t<e$ снова 0 и при $t\ge e$ принимает постоянное значение 1.

2.2. Двоичные операторы технических систем

Пусть имеется множество переключательных процессов $x_1(t),\ldots,x_n(t)$. Закон G, по которому это множество преобразуется в переключательный процесс $y(t)$, называется *двоичным оператором.* Таким образом,

$$y(t) = G[x_1(t), ..., x_n(t)]. \qquad (2.5)$$

В технических системах $x_1(t), ..., x_n(t)$ означают входные процессы, $y(t)$ – выходной процесс, а G – оператор системы. Оператор, реализующий преобразование (2.5), называется *n-местным*, по числу преобразуемых процессов. На операторном языке преобразуемые процессы $x_1(t), ..., x_n(t)$ называются воздействиями на оператор *G*, а результирующий процесс $y(t)$ – *реакцией* оператора. Мы ограничимся рассмотрением операторов, удовлетворяющих следующему условию (*принцип физической осуществимости*): значение реакции $y(t)$ в любой момент времени t зависит только от значений воздействий $x_1(t_1), ..., x_n(t_n)$ в предшествующие $t_1, ..., t_n$ или текущий t моменты $(t_1 < t, ..., t_n < t)$ и от значений самой реакций $y(t_*)$ в предшествующие моменты $t_*(t_* < t)$. Если зависимость $y(t)$ от $y_*(t)$ существенна, то оператор называется *оператором с памятью*, если несущественна – *оператором без памяти.* Число моментов $t_{*1}, ..., t_{*s}$ $(t_{*i} < t)$, таких, что значение $y(t)$ существенно зависит от значений $y(t_{*1}), ..., y(t_{*s})$, называется *глубиной памяти* оператора. Это число может быть конечным или бесконечным. В первом случае имеем *оператор с конечной памятью*, во втором – с *бесконечной.* Оператор без памяти называется *временным*, если $y(t)$ существенно зависит от значения воздействий $x_i(t_i)$ в предшествующие моменты времени $t_i(t_i < t)$, *и логическим* – в противном случае, то есть если $y(t)$ зависит только от значений воздействий $x_1(t), ..., x_n(t)$ в тот же текущий момент *t.* Для логического оператора зависимость (2.5) реакции от воздействий конкретизируется

$$y = f(x_1, ..., x_n), \qquad (2.6)$$

где f – некоторая булева функция; $x_1, ..., x_n, y$ – мгновенные значения воздействий и реакции в один и тот же произвольный момент времени *t*.

Двоичный оператор можно задать с помощью уравнения, связывающего значение $y(t)$ со значениями $x_1(t_1), ..., x_n(t_n), y(t_*)$, где $t_i \le t$, $t_* < t$, посредством алгоритма, позволяющего вычислить значения $y(t)$ для любого *t*, и т.д. Удобным способом задания произвольного оператора является его *структурное представление* в виде суперпозиции (схемы) из *элементарных операторов.* Элементарным считается оператор, который является простейшим и потому неделим, то есть не представим суперпозицией более простых операторов. Удобство такого представления в том, что изучение произвольного оператора сводится к изучению существенно более простых элементарных операторов, число которых конечно.

Задачи изучения операторов технических систем можно разделить на три типа.

1. *Задача анализа оператора* заключается в отыскании реакций $y(t)$ заданного оператора на заданные воздействия $x_1(t), ..., x_n(t)$.

2. *Задача синтеза оператора* состоит в построении оператора, преобразующего заданные воздействия $x_1(t), ..., x_n(t)$ в требуемую реакцию $y(t)$. Под построением оператора понимается какое-нибудь конструктивное его задание – абстрактное или структурное (*абстрактный или структурный синтез*).

3. *Задача синтеза воздействий* заключается в отыскании воздействий на оператор $x_1(t), ..., x_n(t)$ по заданным оператору G и его реакции $y(t)$.

2.3. Элементарные операторы

Будем записывать любой переключательный процесс с неуточненным характером участков (импульсов и пауз) в виде

$$x(t) = u(a_1, a_2)\bar{u}(-, a_3)\cdots u^{(-1)^m}(a_{m-1}, a_m), \quad u \in \{0,1\}, \tag{2.7}$$

где $\bar{u}$ – отрицание u, а

$$u^p = \begin{cases} u & \text{при } p = 1; \\ \bar{u} & \text{при } p = -1. \end{cases} \tag{2.8}$$

Рассмотрим несколько элементарных временных операторов.

1. *Оператор* D_τ *задержки на* τ – это одноместный оператор, преобразующий воздействие $x(t)$ вида (2.7) в реакцию

$$y(t) = D_\tau[x(t)] = x(t-\tau) = u(a_1+\tau, a_2+\tau)\bar{u}(-, a_3+\tau)\cdots u^{(-1)^m}(a_{m-1}+\tau, a_m+\tau), \tag{2.9}$$

то есть сдвигающий входной процесс $x(t)$ на постоянное время задержки τ.

2. *Оператор* D_τ^Φ *фильтрации на* τ – одноместный оператор, преобразующий каждый импульс и паузу $u(a_i, a_{i+1})$ воздействия (2.7) в реакцию

$$y(t) = D_\tau^\Phi[u(a_i, a_{i+1})] = \begin{cases} u(a_i+\tau, a_{i+1}+\tau), & a_{i+1} - a_i \ge \tau; \\ \bar{u}, & a_{i+1} - a_i < \tau, \end{cases} \tag{2.10}$$

то есть сдвигающий входной процесс $x(t)$ на время τ и, кроме того, не пропускающий (фильтрующий) изменения $x(t)$, отстоящие друг от друга ближе, чем на τ.

3. *Оператор достройки паузой до* c – одноместный оператор, преобразующий воздействие $x(t)$ вида (2.7) в реакцию (достройка справа, $c > a_m$)

$$y(t) \equiv x_c(t) = \begin{cases} x(t), & u^{(-1)^m} = 0; \\ x(t)0(a_m, c), & u^{(-1)^m} = 1, \end{cases} \tag{2.11}$$

или в реакцию (достройка слева, $c < a_1$)

$$y(t) \equiv {}_c x(t) = \begin{cases} x(t), & u = 0; \\ 0(c, a_1)x(t), & u = 1. \end{cases} \tag{2.12}$$

4. *Оператор достройки импульсом до* c – одноместный оператор, преобразующий воздействие $x(t)$ вида (2.7) в реакцию (достройка справа, $c > a_m$)

$$y(t) \equiv x^c(t) = \begin{cases} x(t), & u^{(-1)^m} = 1; \\ x(t)1(a_m, c), & u^{(-1)^m} = 0, \end{cases} \tag{2.13}$$

или в реакцию (достройка слева, $c < a_1$)

$$y(t) \equiv {}^c x(t) = \begin{cases} x(t), & u = 1; \\ 1(c, a_1)x(t), & u = 0. \end{cases} \tag{2.14}$$

Операторы достройки выполняются раньше других элементарных операторов.

5. *Оператор усечения до* b – одноместный оператор, преобразующий воздействие $x(t)$ (2.7) в реакцию

$$y(t) \equiv x(t \wedge b) = u(a_1 \wedge b, a_2 \wedge b)\overline{u}(-, a_3 \wedge b) \cdots u^{(-1)^m}(a_{m-1} \wedge b, a_m \wedge b) \qquad (2.15)$$

путем взятия конъюнкции НЛ моментов изменения $x(t)$ с данным моментом b (усечение справа) или в реакцию

$$y(t) \equiv x(t \vee b) = u(a_1 \vee b, a_2 \vee b)\overline{u}(-, a_3 \vee b) \cdots u^{(-1)^m}(a_{m-1} \vee b, a_m \vee b) \qquad (2.16)$$

путем взятия дизъюнкции НЛ данных моментов (усечение слева). Процесс $x(t \wedge b)$ отличается от процесса $x(t)$ заменой на интервале $b < t < \infty$ всех значений $x(t)$ конечным значением. Процесс $x(t \vee b)$ отличается от $x(t)$ заменой при $-\infty < t < b$ всех значений $x(t)$ начальным значением.

6. *Оператор умножения* – двухместный оператор, преобразующий пару воздействий $x_1(t),\ x_2(t)$, не пересекающихся во времени $\left(b_{x_1} \le a_{x_2}\right)$ и таких, что конечное значение первого процесса $x_1(t)$ совпадает с начальным значением второго $x_2(t),$ в реакцию вида

$$y(t) = \begin{cases} x_1(t), & t < b_{x_1}; \\ x_2(t), & t \ge b_{x_1}. \end{cases} \qquad (2.17)$$

Эта реакция называется *произведением процесса* $x_1(t)$ на $x_2(t)$ и обозначается

$$y(t) = x_1(t) \circ x_2(t). \qquad (2.18)$$

Из (2.17) видно, что произведение процесса $x_1(t)$ на $x_2(t)$ до момента b_{x_1} окончания $x_1(t)$ совпадает с $x_1(t),$ с момента a_{x_2} начала $x_2(t)$ совпадает с $x_2(t),$ в интервале $\left[b_{x_1}, a_{x_2}\right]$ равно конечному значению $x_1(t)$ (начальному значению $x_2(t)$). Оператор умножения подчиняется ассоциативному закону, то есть при $b_{x_1} \le a_{x_2}\, b_{x_2} \le a_{x_3}$

$$[x_1(t) \circ x_2(t)] \circ x_3(t) = x_1(t) \circ [x_2(t) \circ x_3(t)] = x_1(t) \circ x_2(t) \circ x_3(t), \qquad (2.19)$$

но не подчиняется коммутативному закону, то есть в общем случае $x_1(t) \circ x_2(t)$ не совпадает с $x_2(t) \circ x_1(t).$

7. *Оператор разбиения* – одноместный оператор, который разбивает процесс $x(t)$ вида (2.7) на два последовательных подпроцесса

$$\left.\begin{aligned} &x_1(t) = u(a_1, a_2)\overline{u}(-, a_3) ... \widetilde{u}(a_{i-1}, a_i), \quad \text{где } \widetilde{u} = u \text{ или } \overline{u}; \\ &x_2(t) = \overline{\widetilde{u}}(a_i, a_{i+1})\widetilde{u}(-, a_{i+2}) ... u^{(-1)^m}(a_{m-1}, a_m), \end{aligned}\right\} \qquad (2.20)$$

так что

$$x(t) = \begin{cases} x_1(t), & t < a_i; \\ x_2(t), & t \ge a_i. \end{cases} \qquad (2.21)$$

Сравнение (2.21) с (2.17) показывает, что

$$x(t) = x_1(t) \circ x_2(t), \qquad (2.22)$$

то есть перемножение подпроцессов $x_1(t)$ и $x_2(t)$ снова дает исходный процесс $x(t)$. Потому операторы умножения и разбиения взаимно обратны. Заключительное изменение в первом подпроцессе $x_1(t)$ разбиения (2.20) назовем точкой деления разбиваемого процесса $x(t)$. Точка деления имеет вид $1'_{a_i}$ или $0'_{a_i}$.

Рассмотрим несколько элементарных логических операторов. Согласно (2.6) такой оператор можно задать с помощью булевой функции, преобразующей мгновенное значение воздействий в любой момент t в мгновенное значение реакции, относящееся к тому же моменту.

1. Конъюнктор – двухместный оператор, преобразующий воздействия $x_1(t)$, $x_2(t)$ в реакцию $y(t)$ согласно конъюнкции (1.1) [1]

$$y = x_1 \wedge x_2 . \tag{2.23}$$

2. Дизъюнктор – двухместный оператор, преобразующий воздействия $x_1(t)$, $x_2(t)$ в реакцию $y(t)$ согласно дизъюнкции (1.2) [1]

$$y = x_1 \vee x_2 . \tag{2.24}$$

3. Инвертор – одноместный оператор, преобразующий воздействие $x(t)$ в реакцию $y(t)$ согласно булевой функции отрицания (1.3) [1]

$$y = \overline{x} . \tag{2.25}$$

4. Дизъюнктивный инвертор (оператор Вебба) – двухместный оператор, преобразующий воздействия $x_1(t), x_2(t)$ в реакцию $y(t)$ согласно булевой функции «отрицание дизъюнкции»

$$y = \overline{x_1 \vee x_2} . \tag{2.26}$$

5. Конъюнктивный инвертор (оператор Шеффера) – двухместный оператор, преобразующий воздействия $x_1(t), x_2(t)$ в реакцию $y(t)$ согласно булевой функции «отрицание конъюнкции»

$$y = \overline{x_1 \wedge x_2} . \tag{2.27}$$

Дизъюнктивный и конъюнктивный инверторы, строго говоря, не могут считаться элементарными операторами, так как они являются суперпозицией операторов (2.23) – (2.25). Однако на практике оба используются часто как элементарные операторы.

2.4. Структурное представление операторов без памяти

Удобство структурного представления операторов (см. п. 2.2) делает целесообразной разработку специальной методики перехода от произвольного содержательного описания оператора к его структурному представлению, то есть к схеме, реализующей оператор в виде суперпозиции конечного числа элементарных операторов. Такой переход включает два этапа: первый – от содержательного описания оператора к его математическому описанию; второй – от математического описания оператора к реализующей его схеме. Первый этап неалгоритмичен и выполняется неформально. Рассмотрим второй этап.

Реакция $y(t)$ оператора без памяти в любой момент t зависит от значений воздействий $x_1(t), ..., x_n(t)$ в тот же момент t, а также от их значений $x_1(t_1), ..., x_n(t_n)$ в некоторые предшествующие моменты $t_1, ..., t_n$ (см. п. 2.2).

Будем считать, что число таких предшествующих моментов для каждого воздействия конечно. Тогда зависимость реакции оператора без памяти от воздействий принимает вид

$$y(t) = f\left[x_1(t), x_1(t_{11}), \ldots, x_1\left(t_{1m_1}\right); \ldots; x_n(t), x_n(t_{n1}), \ldots, x_n\left(t_{nm_n}\right)\right], \tag{2.28}$$

где f – некоторая булева функция; t – текущий момент времени; $t_{ij}\,(t_{ij} < t)$ – предшествующие t моменты, значения воздействий в которых влияют на значение реакции в текущий момент t.

В частном случае, когда значения воздействий в предшествующие моменты несущественны, то есть когда оператор логический, зависимость реакции от воздействий приобретает знакомый вид (2.6)

$$y = f(x_1, \ldots, x_n). \tag{2.29}$$

Здесь $x_1, \ldots, x_n, y$ – мгновенные значения воздействий и реакции, взятые в один и тот же произвольный момент времени. Формулы (2.28), (2.29) дают математическое описание двух типов оператора без памяти: временного и логического.

Начнем с задачи структурного представления логического оператора, то есть построения схемы, реализующей булеву функцию f (2.29) в виде суперпозиции элементарных операторов f_i. При этом достаточно ограничиться только логическими операторами f_i. Набор f_i, позволяющий реализовать любую функцию f, называется *функционально полным* или *базисом.* Образуют базис, например, следующие наборы: 1) конъюнктор и инвертор; 2) дизъюнктор и инвертор; 3) конъюнктор, дизъюнктор и инвертор; 4) оператор Вебба; 5) оператор Шеффера. Будем реализовывать логический оператор в базисе 3. Для такой реализации нужно: 1) перейти от имеющегося представления оператора к соответствующей булевой функции f; 2) привести f к эквивалентному выражению в ДНФ или КНФ (см. п. 1.1 [1]); 3) полученное выражение разложить по элементарным операциям – двухместным конъюнкции и дизъюнкции, используя для этого сочетательный закон (1.10) [1]; 4) сопоставить каждой элементарной операции – конъюнкции, дизъюнкции и отрицанию – соответствующий элементарный логический оператор. При необходимости между этапами 2 и 3 можно выполнить этап упрощения функции f. Для этого выражение f подвергается подходящим эквивалентным преобразованиям (см. п. 1.2 [1]).

Пример 2.1. Реализуем логический оператор, для которого зависимость реакции y от воздействий x_1, x_2, x_3 такова, что $y = 1$ на наборах воздействий 000 и 111. Этап 1) уже выполнен – функция $y = f(x_1, x_2, x_3)$ задана перечислением единичных наборов. 2) Согласно процедуре п. 1.2 ДНФ функции такова: $y = \bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1x_2x_3$. 3) По закону (1.10) получаем $y = (\bar{x}_1\bar{x}_2)\bar{x}_3 \vee (x_1x_2)x_3$. 4) Сопоставляем отрицаниям инверторы, конъюнкциям – конъюнкторы, дизъюнкции – дизъюнктор. В результате находим схему, реализующую оператор (рис. 2.1).

Итак, любой логический оператор можно представить структурно в виде логической схемы, построенной из элементарных логических операторов.

Перейдем к структурному представлению временного оператора. Построим схему, реализующую зависимость (2.28). Введем замену

$$\left.\begin{array}{l}
x_1(t_{11}) = x_{n+1}(t), \ldots, x_1\left(t_{1m_1}\right) = x_{n+m_1}(t), \\
x_2(t_{21}) = x_{n+m_1+1}(t), \ldots, x_2\left(t_{2m_2}\right) = x_{n+m_1+m_2}(t), \\
\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
x_n(t_{n1}) = x_{n+\sum_{i=1}^{n-1} m_i + 1}(t), \ldots, x_n\left(t_{nm_n}\right) = x_{n+\sum_{i=1}^{n} m_i}(t).
\end{array}\right\} \tag{2.30}$$

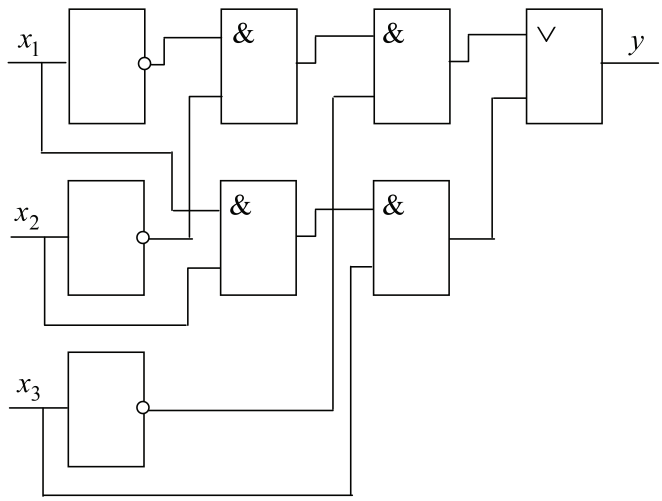

Рис. 2.1

Тогда зависимость (2.28) примет вид булевой функции

$$y = f\left(x_1, ..., x_n, x_{n+1}, ..., x_{n+\sum_{i=1}^{n} m_i}\right) \tag{2.31}$$

от расширенного множества аргументов $x_1,...,x_{n+\sum_{i=1}^{n} m_i}$, в которой x_i и y – мгновенные значения воздействий и реакции, взятые в один и тот же произвольный момент времени. Функция (2.31) имеет тип (2.29), то есть задает некоторый логический оператор. Таким образом, структурное представление временного оператора распадается на структурное представление логического оператора и соотношений (2.30). Первая задача рассмотрена выше. Рассмотрим вторую задачу.

Обратимся, например, к первому соотношению (2.30). Учитывая, что $t_{11} < t$, то есть $t_{11} = t - \tau_{11}$, где $\tau_{11} > 0$, запишем его так: $x_{n+1}(t) = x_1(t - \tau_{11})$ или, используя оператор задержки D_τ,

$$x_{n+1}(t) = D_{\tau_{11}}[x_1(t)]. \tag{2.32}$$

Видим, что любое соотношение (2.30) реализуется при помощи оператора задержки $D_{\tau_{ij}}$ с нужным временем задержки τ_{ij}. При этом для реализации всех соотношений (2.30) нет нужды использовать соответствующее число операторов задержки. Действительно, соединяя последовательно несколько операторов задержки $D_{\tau_1}, ..., D_{\tau_p}$, получаем новый оператор D_τ с суммарным временем задержки $\tau = \sum_{i=1}^{p} \tau_i$. Поэтому достаточно выбрать в качестве элементарного оператор D_τ с временем задержки τ – общим делителем всех времен $\tau_{ij} = t - t_{ij}$ в (2.30). Тогда реализация любого соотношения (2.30) сведется к последовательному соединению нужного числа элементарных операторов D_τ.

Итак, любой временной оператор можно представить структурно в виде логической схемы, построенной из элементарных логических операторов и элементарного оператора задержки.

2.5. Математическая модель надежности системы

Рассмотрим произвольную *систему* (техническую, экономическую, биологическую и т.д.), состоящую из N взаимодействующих подсистем, которые назовем *блоками.* В системе имеется n *входов* и r *выходов.* По входам система получает предусмотренные условиями ее работы полезные воздействия (физические входные сигналы, задачи, подлежащие решению, управляющие команды и т.д.) или вредные воздействия (помехи, вибрация, повышенная температура, влажность и т.д.), влияющие на ее надежность, причем каждый вход предназначен для воздействий одного типа. С выходов системы снимаются различные результаты ее работы (обработанные сигналы, решенные задачи, выполненные команды и т.д.), причем каждый выход характеризует какую-то одну функцию (один результат работы) системы.

Зададим *надежностное состояние* (**НС**) системы двоичным вектором

$$\boldsymbol{y} = (y_1, ..., y_r), \quad y_i \in \{0, 1\}, \tag{2.33}$$

i-я компонента которого y_i характеризует НС i-го выхода системы

$$y_i = \begin{cases} 1, & \text{если система } \textit{работоспособна} \text{ по } i\text{-й функции;} \\ 0, & \text{если система } \textit{неработоспособна} \text{ по } i\text{-й функции} \\ & (\text{частичный отказ } i\text{-го типа}). \end{cases} \tag{2.34}$$

Аналогично зададим НС совокупности блоков двоичным вектором

$$\boldsymbol{a} = (a_1, ..., a_N), \quad a_i \in \{0, 1\}, \tag{2.35}$$

i-я компонента которого a_i характеризует НС i-го блока

$$a_i = \begin{cases} 1, & \text{если } i\text{-й блок работоспособен;} \\ 0, & \text{если } i\text{-й блок отказал.} \end{cases} \tag{2.36}$$

Опишем НС совокупности входов системы вектором

$$\boldsymbol{x} = (x_1, ..., x_n), \quad x_i \in \{0, 1\}, \tag{2.37}$$

i-я компонента которого x_i $(i = 1, ..., n)$ характеризует НС i-го входа

$$a_i = \begin{cases} 1, & \text{если система воспринимает воздействие } i\text{-го типа;} \\ 0, & \text{в противном случае.} \end{cases} \tag{2.38}$$

Описание входов системы при помощи двоичного вектора (2.37) годится и в более общем случае, когда существен не только факт наличия (отсутствия) воздействия каждого типа, но и значения воздействия. При этом множество возможных значений воздействия каждого типа i дискретизуется (если эти воздействия непрерывные) и кодируется двоичным кодом $x_{i_1}, ..., x_{im_i}$; последний заменяет x_i в основном коде (2.37).

Итак, надежностную ситуацию в системе в произвольный момент времени t можно полностью описать тройкой векторов

$$\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{y}), \tag{2.39}$$

где $\boldsymbol{x}$ – НС входов; $\boldsymbol{a}$ – НС блоков; $\boldsymbol{y}$ – НС выходов системы в момент *t*. Это описание – статическое, относящееся к выбранному моменту времени. Реально все три вектора зависят от времени и надежностную эволюцию системы можно описать вектор-функцией

$$\boldsymbol{z}(t) = [\boldsymbol{x}(t), \boldsymbol{a}(t), \boldsymbol{y}(t)]. \quad (2.40)$$

Это описание динамическое, оно охватывает необходимый интервал времени функционирования системы.

Первая компонента в (2.40) – вектор-функция $\boldsymbol{x}(t) = [x_1(t), ..., x_n(t)]$ – задает эволюцию НС входов системы, то есть воздействия на входах системы. Здесь $x_i(t)$ – двоичная функция непрерывного времени *t*, описывающая эволюцию НС *i*-го входа, то есть воздействие на *i*-м входе системы; $x_i(t)$ имеет вид последовательности интервалов наличия и отсутствия *i*-го внешнего фактора, влияющего на надежность системы. Из физического смысла функции $x_i(t)$ следует, что она определена в любой момент бесконечного временного интервала $t\,(-\infty < t < \infty)$, причем на любом конечном подынтервале этого интервала $x_i(t)$ изменяется конечное число раз. Условимся, что значение функции $x_i(t)$ в момент ее изменения $t = a$ совпадает с ее значением при $t > a$. Таким образом, воздействия на входы системы $x_1(t), ..., x_n(t)$ – некоторые переключательные процессы.

Вторая компонента в (2.40) – вектор-функция $\boldsymbol{a}(t) = [a_1(t), ..., a_N(t)]$ – задает эволюцию НС блоков системы, причем $a_i(t)$ – двоичная функция времени, задающая эволюцию НС *i*-го блока в виде последовательности интервалов наличия и отсутствия работоспособности блока. Аналогично предыдущему убеждаемся, что процессы надежностной эволюции блоков $a_1(t), ..., a_N(t)$ – переключательные процессы. Назовем их *надежностными процессами* (**НП**) *в блоках.*

Третья компонента в (2.40) – вектор-функция $\boldsymbol{y}(t) = [y_1(t), ..., y_r(t)]$ – описывает эволюцию НС выходов системы, то есть эволюцию работоспособности системы в отношении ее функций. Здесь $y_i(t)$ – двоичный процесс, задающий эволюцию НС *i*-го выхода, то есть эволюцию работоспособности системы в отношении ее *i*-й функции; $y_i(t)$ имеет вид последовательности интервалов выполнения и невыполнения функции. Как и выше, устанавливаем, что процессы надежностной эволюции выходов системы $y_1(t), ..., y_r(t)$ – переключательные процессы. Назовем их НП *на выходах системы.*

Итак, надежностную эволюцию в системе можно полностью описать тремя группами переключательных процессов:

1) воздействия $x_1(t), ..., x_n(t)$ на n входов системы, влияющие на ее надежность;

2) НП $a_1(t), ..., a_N(t)$ в N блоках системы;

3) НП $y_1(t), ..., y_r(t)$ на r выходах системы, характеризующие эволюцию работоспособности в отношении r различных функций системы.

Эти группы процессов зависимы. Действительно, выполнение системой возложенных на нее функций определяется НП в блоках системы и входными воздействиями на систему. Из физических соображений следует, что выполнение системой любой *i*-й функции в любой момент времени t зависит только от значений НП в блоках и значений входных воздействий в тот же момент t и предшествующие моменты (и, возможно, от выполнения системой ее функций в предшествующие моменты времени). Таким образом,

$$\left.\begin{aligned} &y_1(t) = G_1[x_1(t),...,x_n(t);\, a_1(t),\, ...,\, a_N(t)], \\ &\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ &y_r(t) = G_r[x_1(t),...,x_n(t);\, a_1(t),\, ...,\, a_N(t)], \end{aligned}\right\} \qquad (2.41)$$

где G_i $(i = 1, ..., r)$ – некоторые двоичные операторы, удовлетворяющие принципу физической осуществимости. Эти операторы назовем собственными *надежностными операторами* (**НО**) системы. Совокупность собственных НО системы

$$\boldsymbol{G} = (G_1, ..., G_r) \qquad (2.42)$$

является наиболее полной *надежностной* характеристикой системы. Зная эту характеристику, можно из соотношений (2.41) вычислить НП $y_1(t), ..., y_r(t)$ на выходах системы при любых заданных входных воздействиях $x_1(t), ..., x_n(t)$ и НП $a_1(t), ..., a_N(t)$ в блоках системы. Получаемые НП $y_1(t), ..., y_r(t)$ полностью характеризуют надежность результатов работы системы. По ним, в частности, можно вычислить любой *показатель надежности* (**ПН**) системы. Действительно, это связано с тем, что каждый ПН R представляет собой некоторый функционал F от $y_1(t), ..., y_r(t)$

$$R = F[y_1(t), ..., y_r(t)]. \qquad (2.43)$$

Рассмотрим этот вопрос подробнее. Выбор того или иного ПН системы зависит от назначения системы и *надежностного режима* ее работы – *без восстановления* или *с восстановлением* отказавших блоков. Для системы без восстановления основным ПН является *наработка Т до отказа*, определяемая как интервал времени от момента начала эксплуатации системы t_0 до ее первого отказа. Другим ПН этих систем может служить *функция готовности* $K_г(t)$, определяемая как

$$K_г(t) = \begin{cases} 1, & \text{если система в момент } t \text{ работоспособна;} \\ 0, & \text{если система в момент } t \text{ неработоспособна,} \end{cases} \qquad (2.44)$$

и *функция надежности* $P(t)$

$$P(t) = \begin{cases} 1, & \text{при отсутствии отказов на интервале } [t_0, \mathrm{t}); \\ 0, & \text{в противном случае.} \end{cases} \qquad (2.45)$$

Функции $K_г(t)$, $P(t)$ невосстанавливаемой системы являются переключательными процессами одинакового вида

$$K_г(t) = P(t) = 0'_T\,, \qquad (2.46)$$

так что ПН $K_г(t)$, $P(t)$, T оказываются зависимыми. Для систем с восстановлением основными ПН служат функция готовности $K_г(t)$ (имеющая в отличие от (2.46) вид переключательного процесса с несколькими изменениями) и ресурс V, определяемый как интервал времени от момента начала эксплуатации системы t_0 до момента ее окончательного (невосстанавливаемого) отказа. Используется и функция надежности $P(t)$, имеющая вид (2.46), а также *коэффициент готовности* $K_г$ – доля времени, в течение которого система работоспособна. Он равен

$$K_г = \left[\int_{t_0}^{t_0+V} K_г(t)dt\right] \Big/ V\,, \qquad (2.47)$$

то есть является средним значением функции готовности $K_г(t)$ на интервале (t_0, t_0+V). Часто надежность восстанавливаемой системы характеризуют *наработкой* T_i *между отказами*, определяемой как интервал времени от момента очередного i-го восстановления системы до момента следующего после него отказа, и временем i-го восстановления $T_{вi}$. Как видно из (2.45), (2.47), готовность $K_г(t)$ является первичным ПН системы, через который выражаются другие ее ПН. Отметим, что при $T_i = T$, $T_{вi} = T_в$

$$K_г = T/(T+T_в)\,. \tag{2.48}$$

Вычисление ПН по соотношению (2.43) требует знания *критерия отказа* системы. Этот критерий зависит от назначения системы, режима эксплуатации и т.д. Если по условиям работы система должна выполнять одновременно все r своих функций, то критерием отказа системы является невыполнение хотя бы одной из этих функций (случай 1). Если система должна выполнять, по крайней мере, одну из возможных функций, то критерий отказа – невыполнение всех r функций (случай 2). Если система должна выполнять не менее $p\ (1<p<r)$ функций (безразлично каких), то критерий отказа системы – невыполнение не менее $r-p$ каких-либо функций (случай 3). Возможны и более сложные критерии отказа системы, например, учитывающие неравноценность различных функций системы. Знание критерия отказа системы позволяет выразить ее ПН $K_г(t)$ и $P(t)$ через НП на выходах системы $y_1(t), \ldots, y_r(t)$

$$K_г(t) = y_{экв}(t) = \left|\begin{array}{ll} y(t) & \text{(однофункциональная система);} \\ \left.\begin{array}{ll} \bigwedge\limits_{i=1}^{r} y_i(t) & \text{в случае 1,} \\ \bigvee\limits_{i=1}^{r} y_i(t) & \text{в случае 2,} \\ \bigvee\limits_{s=p}^{r} \bigvee\limits_{i_1 \neq \ldots \neq i_s} \left[y_{i_1}(t) \ldots y_{i_s}(t)\right] & \text{в случае 3.} \end{array}\right] & \text{(многофункциональная система).} \end{array}\right. \tag{2.49}$$

Здесь $\wedge$ и $\vee$ – конъюнкция и дизъюнкция двузначной логики, а $y_{экв}(t)$ – эквивалентный НП в системе, полученный объединением всех НП на выходах системы

$$P(t) = \begin{cases} 1, & \text{если } y_{экв}(\tau) = 1 \text{ при } 0 \le \tau \le t; \\ 0, & \text{в противном случае.} \end{cases} \tag{2.50}$$

Таким образом, вычисление различных ПН системы сводится к одной более общей задаче – определению НП на выходах системы.

Введенные выше операторные зависимости (2.41) НП на выходах произвольной системы от НП на ее входах и в блоках задают *надежностную модель системы.* Эта модель имеет две важные особенности: 1) работоспособность системы определяется не только работоспособностью ее блоков, но и воздействиями на ее входах; 2) работоспособность системы в любой текущий момент времени может зависеть от работоспособности блоков и входных воздействий не только в этот, но и в предшествующие моменты (и, возможно, от предшествующих значений работоспособности системы).

Заключение

С математической точки зрения введенная надежностная модель системы в виде операторной зависимости (2.41) замечательна тем, что ее структурным воплощением оказывается некоторый динамический автомат (типа рис. 2.1), входные процессы которого связаны с его выходными процессами указанной зависимостью. Таким образом, вычисление НП на выходах системы по известным НП в ее блоках и на входах сводится к хорошо известным и детально разработанным в теории автоматов методам вычисления выходных процессов динамических автоматов по их входным процессам. Поскольку в статике в любой фиксированный момент времени выходные значения автомата связаны с его входными значениями суперпозицией операций двузначной логики, а в динамике выходные процессы автомата связаны с его входными процессами суперпозицией операций НЛ, можно говорить, что предложенная модель и вытекающие из нее теория и методы расчета надежности систем являются логическими.

Список литературы

1. Левин, В.И. Логические методы в теории надежности. I. Математический аппарат / В.И. Левин // Вестн. Тамб. гос. техн. ун-та. – 2009. – Т. 15, № 4. – С. 873–884.

2. Левин, В.И. Динамика конечных автоматов и надежность сложных систем / В.И. Левин // Автоматика и вычисл. техника. – 1976. – № 6. – С. 17–24.

3. Левин, В.И. Введение в динамическую теорию конечных автоматов / В.И. Левин. – Рига : Зинатне, 1975. – 376 с.

Logical Methods in Reliability Theory. II. Mathematical Models of Reliability

V.I. Levin

Penza State Technological Academy;
levin@pgta.ac.ru

Key words and phrases: binary operator; dynamic machine; logic theory of reliability; operator structure; reliability process; switching process.

Abstract: The paper proposes machine logical model of system reliability. The input processes of the machine model reliability processes in system blocks, and output process of the machine model reliability process of the system.

Logische Methoden in der Sicherheitstheorie. II. Matematische Modelle der Sicherheit

Zusammenfassung: Es ist das autonomlogische Modell der Systemsicherheit vorgeschlagen. In diesem Modell modellieren die Eingangsprozesse des Automates die Sicherprozesse in den Systemblöcke, und die Ausgangsprozesse des Automates – die Sicherprozesse im System.

Méthodes logiques dans la théorie de défaillance.
II. Modèles mathématiques de défaillance

Résumé: Est proposé le modèle automatique logique des défaillances des systèmes. Les processus d'entrée y modèlent les processus fiables dans les blocs du système et les processus de sorti de l'automate – les processus fiables dans le système-même.

Автор: ***Левин Виталий Ильич*** – доктор технических наук, профессор, заведующий кафедрой «Научные технологии», ГОУ ВПО «ПГТА».

Рецензент: ***Муромцев Дмитрий Юрьевич*** – доктор технических наук, профессор, заведующий кафедрой «Конструирование радиоэлектронных и микропроцессорных систем», ГОУ ВПО «ТГТУ».
