Д. т. н. В. И. ЛЕВИН

Россия, Пензенская государственная технологическая академия
E-mail: levin@pgta.ru

Дата поступления в редакцию
30.03 2010 г.

Оппонент *к. т. н. В. Е. ПАТРАЕВ*
(ОАО «Информационные спутниковые системы», г. Железногорск)

ЛОГИЧЕСКИЕ МЕТОДЫ РАСЧЕТА НАДЕЖНОСТИ СИСТЕМ

Предложена автоматно-логическая модель надежности систем, в которой входные процессы автомата моделируют надежностные процессы в блоках системы, а выходные — надежностные процессы в самой системе.

При проектировании технических систем их обычно рассчитывают на надежность. Имеется множество источников, указывающих, как рассчитать нужную вероятностную характеристику (показатель) надежности системы по аналогичным характеристикам ее элементов. В качестве характеристик надежности обычно выбирают среднее время $T_{\text{ср}}$ безотказной работы или вероятность $P(t)$ безотказной работы за время t. Но вероятностные характеристики — не первичные величины, а результат их усреднения. Поэтому вероятностный расчет надежности системы не элементарен и для сколько-нибудь сложных систем сопряжен с большими трудностями. Кроме того, вероятностные характеристики надежности элементов не всегда могут быть определены из-за большой трудоемкости необходимых статистических испытаний. Наконец, разработка новых систем может требовать рассмотрения новых вероятностных характеристик надежности. В этих условиях кажется целесообразным построение такой теории и методов расчета надежности систем, где оперируют первичными (которые невозможно вывести из других) величинами, относящимися к надежности системы и ее элементов, и устанавливают связь между ними. Действительно, при таком подходе расчет надежности систем становится элементарным, что должно расширить класс анализируемых систем, отпадает необходимость в специальном изучении конкретных характеристик надежности систем, т. к. их всегда можно выразить через указанные первичные величины. К тому же, естественно ожидать, что опытное определение первичных характеристик надежности элементов проще, чем вероятностных.

В предлагаемой в данной работе теории первичными считаются последовательные моменты t_i отказов и восстановлений блоков (элементов) и аналогичные моменты T_k для системы в целом, а задача состоит в отыскании зависимостей $T_k=f_k(t_i)$ между ними. Эта теория обладает не только всеми указанными выше достоинствами, но к тому же имеет и другие. Рассмотрим их.

1. Набор функций $\{f_k\}$ является наиболее полной качественной информацией о влиянии надежности блоков на надежность системы, т. к. по нему можно количественно предсказать надежностное поведение системы при любых значениях надежности ее блоков (в частности, вычислить те или иные характеристики надежности системы).

2. Сравнивать надежность систем можно даже в отсутствие числовых данных о надежности блоков. Равнонадежным системам соответствуют эквивалентные наборы $\{f_k\}$, более надежной системе соответствуют бóльшие значения функций, выражающих моменты отказов, и меньшие значения функций, выражающих моменты восстановлений.

3. Можно изучать не только модели обычных систем, в которых надежностное состояние системы в целом в любой момент времени полностью определяется надежностными состояниями ее блоков в тот же момент, но и модели систем с запоминанием предыдущих состояний.

4. Зная набор функций $\{f_k\}$, можно, наблюдая состояния блоков системы, прогнозировать ее индивидуальное надежностное поведение и на этой основе организовать рациональное техническое обслуживание.

В качестве модели надежности системы был выбран динамический автомат, на входы которого подаются надежностные процессы в блоках (импульс процесса определяет интервал работоспособности блока, пауза — интервал неработоспособности), а с выхода снимается аналогичный процесс для системы. (Надежностным процессом будем называть процесс последовательного изменения состояний системы, определяющий ее надежность). При этом оказалось, что функции влияния f_k всегда выражаются суперпозицией операций непрерывной логики, а это позволяет говорить о применимости логической теории надежности. Эта теория, помимо ее важного самостоятельного значения, полезна и при расчетах надежности по традиционной вероятностной теории, ибо, устанавливая логическую связь между первичными надежностными характеристиками системы и ее блоков, она облегчает расчет и компьютерное моделирование вероятностных характеристик надежности сложных систем, сводя его к известной задаче — отысканию (моделированию) распределения детерминированных функций от случайных аргументов.

Переключательные процессы

Рассмотрим произвольную двоичную функцию непрерывного времени t, генерируемую в некоторой системе (в частности, в технической), т. е. функцию $x(t)$, значения которой принадлежат множеству $\{0, 1\}$. Пусть эта функция удовлетворяет трем условиям: 1) значение функции x в момент ее изменения a 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 t; 0 — постоянный процесс, равный нулю на некотором интервале времени t; $1'$ — изменение значения процесса вида $0\to1$; $0'$ — изменение значения процесса вида $1\to0$; $0'_a$ — изменение $0'$ в момент a; $1'_a$ — изменение $1'$ в момент a; $1(a, b)$ — импульс $1'_a0'_b$; $0(a, b)$ — пауза $0'_a1'_b$. В соответствии с условием 1, в некоторой окрестности момента a изменения значения процесса определяются выражениями

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

Согласно (1), импульс — это интервал единичных значений процесса, включающий начало и не включающий конец, а пауза — интервал нулевых значений процесса с аналогичными включениями:

$$\begin{aligned}&1(a,b)=\begin{cases}1, & a\le t<b;\\ 0, & t<a \text{ или } t\ge b;\end{cases}\\ &0(a,b)=\begin{cases}0, & a\le t<b;\\ 1, & t<a \text{ или } t\ge b.\end{cases}\end{aligned}\tag{2}$$

Формулы (2) при $a=b$ принимают вид

$$1(a,\ a)\equiv0;\ 0(a,\ a)\equiv1,\tag{3}$$

откуда видно, что импульс (пауза) с совмещенными началом и концом фактически есть отсутствие импульса (паузы), т. е. это вырожденный участок, и он может быть исключен из рассмотрения. Однако при этом из (3) следует возможность формально рассматривать отсутствие импульса (паузы), т. е. тождественный 0 (тождественную 1), как импульс (паузу) с совмещенными началом и концом, что часто бывает полезным. Отметим также возможность рассматривать изменения процесса (1) как импульс (паузу) на бесконечном интервале:

$$1'_a=1(a,\ \infty)=0(-\infty,\ a);\ 0'_a=0(a,\ \infty)=1(-\infty,\ a).\tag{4}$$

Введем необходимые определения. Пусть $x(t)$ — любой переключательный процесс, отличный от тождественного 0 или 1; a_x — момент начала первого изменения и b_x — момент окончания последнего изменения $x(t)$. 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_∞ при $t>b_x$ назовем конечным значением процесса, 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 $L\le1$ процесс называется простым, при $L\ge2$ — сложным. Два переключательных процесса равны, если у них одинаковое число соответственно однотипных изменений, моменты которых совпадают. Два переключательных процесса считаем эквивалентными, если при любой численной конкретизации буквенных обозначений 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 будем опускать начальное и конечное постоянные значения, а моменты промежуточных изменений указывать один раз либо в импульсе, либо в соседней паузе. Например, один и тот же процесс можно записать в виде $x(t)=1'_a0'_b1'_c0'_d1'_e$ 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.

Двоичные операторы технических систем

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

$$y(t)=G[x_1(t),\ ...,\ x_n(t)].\tag{5}$$

В технических системах $x_1(t)$, ..., $x_n(t)$ означают входные процессы, $y(t)$ — выходной процесс, G — оператор системы. Оператор, реализующий преобразование (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. Ограничимся рассмотрением операторов, удовлетворяющих следующему условию (принципу 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 реакции $y(t_*)$ в предшествующие моменты t_* ($t_*<t$). Если зависимость $y(t)$ от $y(t_*)$ существенна, 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 зна-

чений $y(t_{*1})$, ..., $y(t_{*s})$, 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 t_i (t_i<t), и логическим — в противном случае, т. е. если $y(t)$ зависит только от значений воздействий $x_1(t)$, ..., $x_n(t)$ в тот же текущий момент t. Для логического оператора зависимость (5) реакции от воздействий конкретизируется:

$$y=f(x_1, \; ..., \; x_n), \tag{6}$$

где f — некоторая булева функция;
x_1, ..., x_n, y — соответственно мгновенные значения воздействий и реакции оператора в 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. Удобным способом задания произвольного оператора является его структурное представление в виде суперпозиции (схемы) из элементарных операторов. Элементарным считается оператор, который является простейшим, т. е. не может быть представлен 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. Задача анализа оператора заключается в отыскании реакций $y(t)$ заданного оператора на заданные воздействия $x_1(t)$, ..., $x_n(t)$. 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). Задача синтеза воздействий заключается в отыскании воздействий $x_1(t)$, ..., $x_n(t)$ на оператор по заданным оператору G и его реакции $y(t)$.

Элементарные операторы

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

$$x(t) = u(a_1, a_2)\bar{u}(-, a_3)...u^{(-1)^m}(a_{m-1}, a_m), \;\; u \in \{0,1\}, \tag{7}$$

причем

$$u^p = \begin{cases} u \;\text{ при }\; p = 1; \\ \bar{u} \;\text{ при }\; p = -1, \end{cases} \tag{8}$$

где $\bar{u}$ — отрицание u;
m–1 — показывает число участков.

Рассмотрим несколько элементарных временных операторов.

1. Оператор D_τ задержки на τ — это одноместный оператор, преобразующий воздействие $x(t)$ вида (7) в реакцию

$$\begin{aligned} & y(t) = D_\tau[x(t)] = x(t-\tau) = \\ & = u(a_1+\tau, \, a_2+\tau)\bar{u}(-, \, a_3+\tau)\ldots \\ & \ldots u^{(-1)^m}(a_{m-1}+\tau, \, a_m+\tau), \end{aligned} \tag{9}$$

т. е. 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 (7) в реакцию

$$\begin{aligned} & y(t) = D_\tau^\Phi[u(a_i, \; a_{i+1})] = \\ & = \begin{cases} u(a_i+\tau, \; a_{i+1}+\tau), \;\; a_{i+1}-a_i \ge \tau; \\ \bar{u}, \;\; a_{i+1}-a_i < \tau, \end{cases} \end{aligned} \tag{10}$$

т. е. 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 (7) в реакцию (достройка справа, $c>a_m$)

$$y(t) \equiv x_c(t) = \begin{cases} x(t), \;\; u^{(-1)^m} = 0; \\ x(t)0(a_m, \; c), \;\; u^{(-1)^m} = 1, \end{cases} \tag{11}$$

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

$$y(t) \equiv {}_c x(t) = \begin{cases} x(t), \;\; u = 0; \\ 0(c, \; a_1)x(t), \;\; u = 1. \end{cases} \tag{12}$$

4. Оператор достройки импульсом до c — одноместный оператор, преобразующий воздействие $x(t)$ вида (7) в реакцию (достройка справа, $c>a_m$)

$$y(t) \equiv x^c(t) = \begin{cases} x(t), \;\; u^{(-1)^m} = 1; \\ x(t)1(a_m, \; c), \;\; u^{(-1)^m} = 0, \end{cases} \tag{13}$$

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

$$y(t) \equiv {}^c x(t) = \begin{cases} x(t), \;\; u = 1; \\ 1(c, \; a_1)x(t), \;\; u = 0. \end{cases} \tag{14}$$

Операторы достройки выполняются раньше других элементарных операторов.

5. Оператор усечения до b — одноместный оператор, преобразующий воздействие $x(t)$ вида (7) путем взятия конъюнкции непрерывной логики моментов изменения $x(t)$ с данным моментом b (усечение справа) в реакцию

$$\begin{aligned} & y(t) \equiv x(t \wedge b) = u(a_1 \wedge b, \;\; a_2 \wedge b)\bar{u}(-, \;\; a_3 \wedge b)\ldots \\ & \ldots u^{(-1)^m}(a_{m-1} \wedge b, \;\; a_m \wedge b) \end{aligned} \tag{15}$$

или путем взятия дизъюнкции непрерывной логики данных моментов (усечение слева) в реакцию

$$y(t) \equiv x(t \vee b) = u(a_1 \vee b,\ a_2 \vee b)\bar{u}(-,\ a_3 \vee b)\ldots$$
$$\ldots u^{(-1)^m}(a_{m-1} \vee b,\ a_m \vee b). \qquad (16)$$

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 ($b_{x_1} \leq a_{x_2}$) и таких, что конечное значение процесса $x_1(t)$ совпадает с начальным значением процесса $x_2(t)$, в реакцию вида

$$y(t) = \begin{cases} x_1(t), & t < b_{x_1}; \\ x_2(t), & t \geq b_{x_1}. \end{cases} \qquad (17)$$

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

$$y(t)=x_1(t)\circ x_2(t). \qquad (18)$$

Из (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 $[b_{x_1}, a_{x_2}]$ — равно конечному значению $x_1(t)$ (начальному значению $x_2(t)$). Оператор умножения подчиняется ассоциативному закону, т. е. при $b_{x_1} \leq a_{x_2} b_{x_2} \leq a_{x_3}$

$$[x_1(t)\circ x_2(t)]\circ x_3(t)=x_1(t)\circ[x_2(t)\circ x_3(t)]=$$
$$=x_1(t)\circ x_2(t)\circ x_3(t), \qquad (19)$$

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

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

$$\left.\begin{array}{l} x_1(t) = u(a_1, a_2)\bar{u}(-, a_3)...\tilde{u}(a_{i-1}, a_i), \\ \qquad \text{где } \tilde{u} = u \text{ или } \bar{u} \\ x_2(t) = \bar{\tilde{u}}(a_i, a_{i+1})\tilde{u}(-, a_{i+2})...u^{(-1)^m}(a_{m-1}, a_m), \end{array}\right\} \qquad (20)$$

при этом

$$x(t) = \begin{cases} x_1(t), & t < a_i; \\ x_2(t), & t \geq a_i. \end{cases} \qquad (21)$$

Сравнение (21) с (17) показывает, что

$$x(t)=x_1(t)\circ x_2(t), \qquad (22)$$

т. е. 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 (20) назовем точкой деления разбиваемого процесса $x(t)$. Точка деления имеет вид $1'_{a_i}$ или $0'_{a_i}$.

Рассмотрим несколько элементарных логических операторов [1]. Согласно (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 булевой конъюнкции

$$y=x_1\wedge x_2. \qquad (23)$$

2. Дизъюнктор — двухместный оператор, преобразующий воздействия $x_1(t)$, $x_2(t)$ в реакцию $y(t)$ согласно булевой дизъюнкции

$$y=x_1\vee x_2. \qquad (24)$$

3. Инвертор — одноместный оператор, преобразующий воздействие $x(t)$ в реакцию $y(t)$ согласно булевой функции отрицания

$$y = \bar{x}. \qquad (25)$$

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

$$y = \overline{x_1 \vee x_2}. \qquad (26)$$

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

$$y = \overline{x_1 \wedge x_2}. \qquad (27)$$

Дизъюнктивный и конъюнктивный инверторы, строго говоря, не могут считаться элементарными операторами, т. к. они являются суперпозицией операторов (23)—(25). Однако на практике оба часто используются как элементарные операторы.

Структурное представление операторов без памяти

Для структурного представления операторов целесообразно разработать специальную методику перехода от произвольного содержательного описания оператора к его структурному представлению, т. е. к схеме, реализующей оператор в виде суперпозиции конечного числа элементарных операторов. Такой переход выполняется в два этапа — от содержательного описания оператора к его математическому описанию и 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 времени 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. Будем считать, что число таких предшествующих моментов для каждого воздействия конечно. Тогда зависимость реакции оператора без памяти от воздействий принимает вид

$$\begin{array}{l} y(t) = f[x_1(t),\ x_1(t_{11}),\ ...,\ x_1(t_{1m_1});\ ...; \\ x_n(t),\ x_n(t_{n1}),\ ...,\ x_n(t_{nm_n})], \end{array} \qquad (28)$$

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, т. е. когда оператор логический, зависимость реакции от воздействий приобретает тот же вид, что и зависимость (6), т. е.

$$y=f(x_1, ..., x_n). \qquad (29)$$

Формулы (28), (29) дают математическое описание двух типов оператора без памяти — временного и логического.

Начнем с задачи структурного представления логического оператора, т. е. построения схемы, реализующей булеву функцию f из (29) в виде суперпозиции нескольких элементарных операторов f_i. При этом достаточно ограничиться только логическими операторами. 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. Чтобы реализовать логический оператор в базисе 3, необходимо выполнить следующее:

— перейти от имеющегося представления оператора к соответствующей булевой функции f;
— привести функцию f к эквивалентному выражению в дизъюнктивной или конъюнктивной или нормальной форме (ДНФ или КНФ) (см. в [1]);
— полученное выражение разложить по элементарным операциям — двухместным конъюнкции и дизъюнкции, используя сочетательный закон;
— каждой элементарной операции — конъюнкции, дизъюнкции и отрицанию — поставить в соответствие элементарный логический оператор.

При необходимости между вторым и третьим этапами можно выполнить этап упрощения функции f. Для этого выражение f подвергается эквивалентным преобразованиям.

Для примера реализуем в приведенной последовательности 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. Поскольку первый этап уже выполнен — функция $y=f(x_1, x_2, x_3)$ задана перечислением единичных наборов, сразу перейдем к определению ДНФ нашей функции: $y=\bar{x}_1\bar{x}_2\bar{x}_3 \vee x_1x_2x_3$. После этого получим $y=(\bar{x}_1\bar{x}_2)\bar{x}_3 \vee (x_1x_2)x_3$. В соответствие отрицаниям поставим инверторы, конъюнкциям — конъюнкторы, дизъюнкции — дизъюнктор, в результате находим схему, реализующую логический оператор (**рис. 1**).

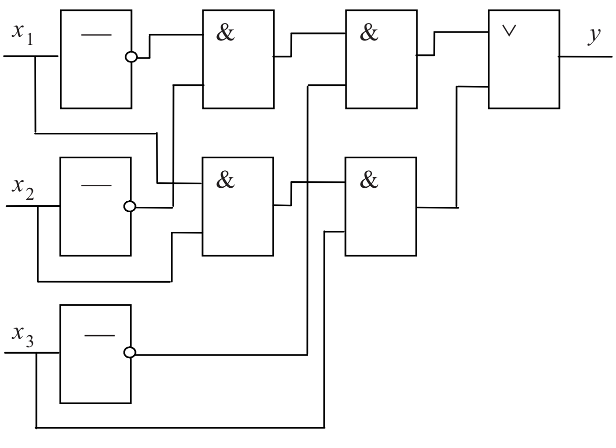

Рис. 1. Схема, реализующая логический оператор

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 (28). Введем замену:

$$\begin{aligned} &x_1(t_{11})=x_{n+1}(t),\ \ldots,\ x_1(t_{1m_1})=x_{n+m_1}(t);\\ &x_2(t_{21})=x_{n+m_1+1}(t),\ \ldots,\ x_2(t_{2m_2})=x_{n+m_1+m_2}(t);\\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ &x_n(t_{n1})=x_{n+\sum_{i=1}^{n-1}m_i+1}(t),\ \ldots,\ x_n(t_{nm_n})=x_{n+\sum_{i=1}^{n}m_i}(t). \end{aligned} \qquad (30)$$

Тогда зависимость (28) примет вид булевой функции от расширенного множества аргументов $x_1, ..., x_{n+\sum_{i=1}^{n}m_i}$, т. е.

$$y=f\left(x_1,\ ...,\ x_n,\ x_{n+1},\ ...,\ x_{n+\sum_{i=1}^{n}m_i}\right), \qquad (31)$$

в которой x_i и y — мгновенные значения воздействий и реакции оператора, взятые в один и тот же, произвольный момент времени. Функция (31) имеет тип (29), т. е. задает некоторый логический оператор.

Таким образом, задача структурного представления временного оператора распадается на две: структурное представление логического оператора и соотношений (30). Первая задача рассмотрена выше. Рассмотрим вторую.

Обратимся, например, к первому соотношению в (30). Учитывая, что $t_{11}<t$, т. е. $t_{11}=t-\tau_{11}$, где $\tau_{11}>0$, запишем его как $x_{n+1}(t)=x_1(t-\tau_{11})$ или, используя оператор задержки D_τ, как

$$x_{n+1}(t)=D_{\tau_{11}}[x_1(t)]. \qquad (32)$$

Очевидно, что любое соотношение (30) реализуется при помощи оператора задержки $D_{\tau_{ij}}$ с нужным временем задержки τ_{ij}. При этом для реализации всех соотношений в (30) нет нужды использовать соответствующее число операторов задержки. Действительно, соединяя последовательно несколько операторов задержки D_{τ_1}, …, D_{τ_p}, получаем новый опертор 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 интервалов времени $\tau_{ij}=t-t_{ij}$ в (30). Тогда реализация любого соотношения в (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.

Математическая модель надежности системы

Рассмотрим модель произвольной системы (это может быть техническая, экономическая, биологическая и т. д. система), состоящей 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

$$\boldsymbol{y}=(y_1,\ ...,\ y_r),\ y_i\in\{0,\ 1\}, \tag{33}$$

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

$$y_i=\begin{cases}1, & \text{если система работоспособна по } i\text{-й функции;}\\ 0, & \text{если система неработоспособна по } i\text{-й функции (частичный отказ } i\text{-го типа).}\end{cases} \tag{34}$$

Аналогично зададим НС совокупности блоков двоичным вектором

$$\boldsymbol{a}=(a_1,\ ...,\ a_N),\ a_i\in\{0,\ 1\}, \tag{35}$$

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

$$a_i=\begin{cases}1, & \text{если } i\text{-й блок работоспособен;}\\ 0, & \text{если } i\text{-й блок отказал.}\end{cases} \tag{36}$$

Опишем НС совокупности входов системы вектором

$$\boldsymbol{x}=(x_1,\ ...,\ x_n),\ x_i\in\{0,\ 1\}, \tag{37}$$

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

$$a_i=\begin{cases}1, & \text{если система воспринимает воздействие } i\text{-го типа;}\\ 0 & \text{в противном случае.}\end{cases} \tag{38}$$

Описание входов системы при помощи двоичного вектора (37) годится и в более общем случае, когда важен не только сам факт наличия (отсутствия) воздействия каждого типа, но и значения этих воздействий. При этом множество возможных значений воздействия каждого типа i дискретизуется (если эти воздействия непрерывные) и кодируется двоичным кодом x_{i_1}, …, x_{im_i}, причем последний заменяет x_i в основном коде (37).

Итак, надежностную ситуацию в системе в произвольный момент времени t можно полностью описать тремя векторами, а именно

$$z=(\boldsymbol{x},\ \boldsymbol{a},\ \boldsymbol{y}), \tag{39}$$

где $\boldsymbol{x}$ — НС входов;
$\boldsymbol{a}$ — НС блоков;
$\boldsymbol{y}$ — НС выходов системы в момент t.

Описание (39) — статическое, относящееся к выбранному моменту времени. Поскольку все три вектора зависят от времени, надежностную эволюцию системы можно описать вектор-функцией

$$z(t)=[\boldsymbol{x}(t),\ \boldsymbol{a}(t),\ \boldsymbol{y}(t)]. \tag{40}$$

Такое описание — динамическое, оно охватывает интервал времени функционирования системы. Первая компонента в (40) — вектор-функция $\boldsymbol{x}(t)=[x_1(t), ..., x_n(t)]$ — задает эволюцию НС входов системы, т. е. воздействия на входах системы. Здесь $x_i(t)$ — двоичная функция непрерывного времени t, описывающая эволюцию НС i-го входа, т. е. воздействие на i-м входе системы. Функция $x_i(t)$ 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 его подынтервале $x_i(t)$ 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), ..., $x_n(t)$ есть некоторые переключательные процессы.

Вторая компонента в (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 случаю, убеждаемся, что процессы надежностной эволюции блоков $a_1(t)$, ..., $a_N(t)$ являются переключательными. Назовем их надежностными процессами (**НП**) в блоках.

Третья компонента в (40) — вектор-функция $\boldsymbol{y}(t)=[y_1(t), ..., y_r(t)]$ — описывает эволюцию НС выходов системы, т. е. эволюцию работоспособности системы в отношении ее функций. Здесь $y_i(t)$ — двоичный процесс, задающий эволюцию НС i-го выхода, т. е. эволюцию работоспособности системы в отношении ее i-й функции; $y_i(t)$ имеет вид последовательности интервалов выполнения и невыполнения функции. Как и раньше, устанавливаем, что процессы надежностной эволюции выходов системы $y_1(t), ..., y_r(t)$ — переключательные. Назовем их надежностными процессами на выходах системы.

Итак, надежностную эволюцию в системе можно полностью описать указанными тремя группами переключательных процессов. Эти группы процессов зависимы. Действительно, выполнение системой возложенных на нее функций определяется НП в блоках системы и входными воздействиями на систему. При этом 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 самый момент t и предшествующие моменты (и, возможно, от выполнения системой ее функций в предшествующие моменты времени). Таким образом,

$$\begin{aligned} &y_1(t) = G_1[x_1(t),\ ...,\ x_n(t);\ a_1(t),\ ...,\ a_N(t)];\\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\ &y_r(t) = G_r[x_1(t),\ ...,\ x_n(t);\ a_1(t),\ ...,\ a_N(t)], \end{aligned} \tag{41}$$

где G_i (i=1, ..., r) — некоторые двоичные операторы, удовлетворяющие принципу физической осуществимости. Назовем эти операторы собственными надежностными операторами (**НО**) системы. Их совокупность

$$\boldsymbol{G}=(G_1,\ ...,\ G_r) \tag{42}$$

является наиболее полной надежностной характеристикой системы. Зная эту характеристику, можно из соотношений (41) вычислить НП на выходах системы при любых заданных входных воздействиях и НП в блоках системы. Получаемые НП $y_1(t), ..., y_r(t)$ полностью характеризуют надежность работы системы. По ним, в частности, можно вычислить любой показатель надежности (**ПН**) системы, поскольку каждый ПН R есть некоторый функционал F от $y_1(t), ..., y_r(t)$:

$$R=F[y_1(t),\ ...,\ y_r(t)]. \tag{43}$$

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 T до отказа, определяемая как интервал времени от момента t_0 начала эксплуатации системы до ее первого отказа. Другим ПН таких систем может служить функция готовности $K_г(t)$, определяемая как

$$K_г(t) = \begin{cases} 1, & \text{если система в момент } t \\ & \text{работоспособна;} \\ 0, & \text{если система в момент } t \\ & \text{неработоспособна,} \end{cases} \tag{44}$$

и функция надежности $P(t)$:

$$P(t) = \begin{cases} 1 & \text{при отсутствии отказов на интервале} \\ & [t_0, t); \\ 0 & \text{в противном случае.} \end{cases} \tag{45}$$

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

$$K_г(t)=P(t)=0'_T, \tag{46}$$

так что ПН $K_г(t)$, $P(t)$, T оказываются взаимозависимыми.

Для систем с восстановлением основными ПН служат функция готовности $K_г(t)$ (имеющая, в отличие от (46), вид переключательного процесса с несколькими изменениями) и ресурс V, определяемый как интервал времени от момента t_0 начала эксплуатации системы до момента ее окончательного (невосстанавливаемого) отказа. Используется и функция надежности $P(t)$, имеющая вид (46), а также коэффициент готовности $K_г$ — доля времени, в течение которого система работоспособна:

$$K_г = \left[\int_{t_0}^{t_0+V} K_г(t)dt \right] \Big/ V, \tag{47}$$

т. е. 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 (45), (47), готовность $K_г(t)$ является первичным ПН системы, через который выражаются другие ее ПН. Отметим, что при T_i=T, $T_{вi}$=$T_в$

$$K_г=T/(T+T_в). \tag{48}$$

Вычисление ПН по соотношению (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 функций, безразлично каких ($1<p<r$), то критерий отказа

— невыполнение не менее $r–p$ каких-либо функций (случай 3).

Возможны и более сложные критерии отказа, учитывающие, например, неравноценность различных функций системы. Знание критерия отказа системы позволяет выразить ее ПН $K_г(t)$ и $P(t)$ через НП на выходах системы $y_1(t)$, ..., $y_r(t)$ следующим образом: для однофункциональной системы

$K_г(t)=y_{экв}(t)=y(t)$;

для многофункциональной системы

$$K_г(t)=y_{экв}(t)=\begin{cases} \bigwedge\limits_{i=1}^{r} y_i(t) & \text{в случае 1,} \\ \bigvee\limits_{i=1}^{r} y_i(t) & \text{в случае 2,} \\ \bigvee\limits_{s=p}^{r} \bigvee\limits_{i_1\neq\ldots\neq i_s} [y_{i_1}(t)\ldots y_{i_s}(t)] & \text{в случае 3.} \end{cases} \quad (49)$$

Здесь $y_{экв}(t)$ — эквивалентный НП в системе, полученный объединением всех НП на выходах системы.

$$P(t)=\begin{cases} 1, & \text{если } y_{экв}(\tau)=1 \text{ при } 0\le\tau\le t, \\ 0 & \text{в противном случае.} \end{cases} \quad (50)$$

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

Введенные выше операторные зависимости (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 внешними 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 достоинство введенной надежностной модели системы в виде операторной зависимости (41) в том, что ее структурным воплощением оказывается некоторый динамический автомат (типа приведенного на рис. 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 непрерывной логики, можно говорить, что предложенная модель и вытекающие из нее теория и методы расчета надежности систем являются логическими.

ИСПОЛЬЗОВАННЫЕ ИСТОЧНИКИ

1. Левин В. И. Логические методы в теории надежности. I. Математический аппарат // Вестник Тамбовского государственного технического университета.— 2009.— Т. 15, № 4.— С. 873—884.

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

3. Левин В. И. Введение в динамическую теорию конечных автоматов.— Рига: Зинатне, 1975.