УДК 629.735.05:681.178:519.713(045)

И.Э. Райчев, А.Г. Харченко

ПРИМЕНЕНИЕ КОНЕЧНЫХ АВТОМАТОВ ДЛЯ РЕАЛИЗАЦИИ АЛГОРИТМОВ КОНТРОЛЯ ПОЛЕТОВ ВОЗДУШНЫХ СУДОВ

Рассмотрены вопросы применения аппарата структурных конечных автоматов для обработки информации параметрических бортовых регистраторов. Исследованы аспекты использования конечных автоматов для реализации алгоритмов контроля полетов. Построены канонические уравнения общего вида, описывающие алгоритмы контроля. Разработана методика построения автоматных моделей для описания алгоритмов контроля полетов воздушных судов.

Полетная информация (ПИ), включающая траекторные параметры, информацию о функционировании систем воздушного судна (ВС) и действиях экипажа по его управлению, широко используется для контроля состояния систем ВС и выявления ошибок в действиях экипажа. Эта информация обрабатывается по алгоритмам контроля на борту ВС или наземными системами после его посадки [1; 2]. Алгоритмы контроля задаются разработчиком ВС и представляют собой сложные логико-алгебраические выражения, содержащие десятки предикатов и логических функций. Так как для каждого типа ВС имеется несколько сотен таких алгоритмов, создание надежного и эффективного программного обеспечения контроля полетов является сложной задачей. Типовое программное обеспечение и альтернативные программные комплексы создаются с использованием подхода прямого программирования алгоритмов контроля полетов [1], что исключает возможность применения средств автоматизации программирования, приводит к высокой вероятности наличия необнаруженных ошибок и усложняет процесс проверки качества программ [3; 4].

Для анализа и реализации алгоритмов контроля полетов предлагается использовать совокупность синтезированных конечных автоматов. Полученные автоматные представления алгоритмов контроля являются эквивалентными исходным. Они могут быть запрограммированы с применением одной из технологий автоматизации программирования для наземной обработки или реализованы в виде аппаратных устройств для обработки ПИ на борту ВС. Автоматные модели проще исходных алгоритмов контроля и могут быть эффективно использованы не только для обработки ПИ, но и для проектирования, кодирования, тестирования и оценки качества программных модулей контроля полетов. В результате проведенных исследований были разработаны два приложения конечных автоматов в системах обработки ПИ.

Первое приложение основывается на возможности использования конечных автоматов для построения принципиальных электрических схем и адаптеров для компьютеров, которые позволят осуществлять копирование ПИ с ленты бортового регистратора, проводить в темпе ввода преобразование дискретной информации кодов параметров в физические величины, записывать эти значения в соответствующие файлы для дальнейшей обработки, выдачи графиков и т.п. Для этого в автоматах должна быть "зашита" структура ленты данного бортового регистратора с расположением параметров по каналам регистрации. Каждая из функций, обеспечивающих этот процесс, может быть реализована с помощью оригинального переключателя (автомата без памяти).

Второе приложение представляет собой использование конечных автоматов для реализации алгоритмов контроля пилотирования и контроля работоспособности ВС.

Как правило, любой конечный структурный автомат для обработки полетной информации составляют из элементарных узлов (элементов), которые реализуют элементарные функции булевской алгебры от одной и двух переменных, сумматор последовательного действия, двоичный счетчик, задержку на один такт и некоторые другие. Из элементов путем соединения их друг с другом составляется схема, определяющая логику функционирования автомата [5].

Рассмотрим разновидность конечных структурных автоматов, которые получают преобразованием логико-алгебраической записи алгоритмов контроля в системы канонических уравнений и которые можно использовать для обработки ПИ. Будем считать, что все элементарные абстрактные автоматы, из которых строится структурный автомат, имеют один и тот же структурный алфавит и работают в одном и том же автоматном времени. Число входных и выходных полюсов, а также внутренних состояний такого автомата конечно. Принципы, на которых будет строиться такая система автоматов, описаны в работах [5; 6].

Алгоритмы контроля пилотирования и работоспособности ВС описывают ситуации контроля и представляются логическими выражениями, состоящими из конечного числа предикатов. В этих выражениях базовыми операциями являются операции конъюнкции, дизъюнкции, отрицания, а также операции их суперпозиции. Значительно упрощается анализ преобразования алгоритмов контроля в связи с отсутствием в них кванторов существования и всеобщности. Предикаты в событиях контроля имеют вид неравенств или равенств, например: $V_{пр} \geqslant 410$, $n_{нд1} \geqslant 28{,}0$, $\alpha_{руд} < 110$, $H_г \geqslant 15$, $H_б \geqslant 7120$ и т.п. В подобных логических функциях участвуют, кроме параметров регистрации, расчетные параметры [2]. Алгоритмы контроля ВС представлены в технической документации, как правило, в виде с.к.н.ф., ранг которых можно понизить, используя законы алгебры логики. Единственным ограничением при преобразовании таких функций является условие сохранения их физического смысла при упрощении (в частности, сохранение определяющих предикатов контроля). После упрощения, осуществив поэтапный анализ алгоритма контроля, можно построить его автоматную модель (синтезировать автомат).

В дальнейшем изложении будем интерпретировать конечный автомат как математическую модель устройства синхронного действия, перерабатывающего дискретную информацию, получаемую автоматом через равноотстоящие друг от друга моменты дискретного (автоматного) времени $t=0,1,2,\ldots,\tau,\ldots$ Модель такого устройства определяется двумя составляющими: логической схемой автомата и оператором, который реализуется схемой [6].

Подавая на вход устройства с m входами и n выходами в момент времени $t=\tau$ входное слово $x_1(\tau),x_2(\tau),\ldots,x_m(\tau)$, на выходе получаем выходное слово $y_1(\tau),y_2(\tau),\ldots,y_n(\tau)$. Поскольку предикаты, используемые в алгоритмах контроля ВС, вырабатывают значение 0 или 1, будем считать, что входные сигналы составляют конечное множество, которое в нашем случае будет состоять из некоторого конечного числа комбинаций нулей и единиц. Очевидно, что в таком случае мы получим конечные алфавиты для входных и выходных полюсов автомата. Для каждого алгоритма контроля будем иметь столько входных полюсов, из скольких предикатов он состоит. Будем считать, что наш автомат будет иметь на выходе один полюс, на котором могут появляться:

0 – данное событие контроля в копии полета не найдено;

1 – событие в копии полета найдено.

В виду того, что выходное слово в автомате, построенном по предикатам алгоритма контроля, состоит из одной буквы, автомат на отрезке времени $[\tau, T]$ перерабатывает систему входных слов:

$$\begin{gathered} x_1(\tau),x_1(\tau+1),\ldots,x_1(T); \\ x_2(\tau),x_2(\tau+1),\ldots,x_2(T); \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ x_m(\tau),x_m(\tau+1),\ldots,x_m(T) \end{gathered}$$

в систему выходных слов:

$$y(\tau),y(\tau+1),\ldots,y(T).$$

Отрезки вида $[\tau, T]$ присутствуют во многих алгоритмах контроля. Например, алгоритм контроля $S096 = (U27 \geqslant 30) \wedge (\Delta\tau \geqslant 30)$ для самолета Ту-154Б (регистратор МСРП-64-2), означает, что выходом за ограничение считается ситуация, когда напряжение сети U27 равно или превышает 30 В на протяжении интервала времени регистрации, равного 30 с и более.

В случае, когда алгоритм контроля "срабатывает" в каждой точке без учета интервала дискретного времени, входные и выходные последовательности "сжимаются" к одной входной и одной выходной точке дискретного времени:

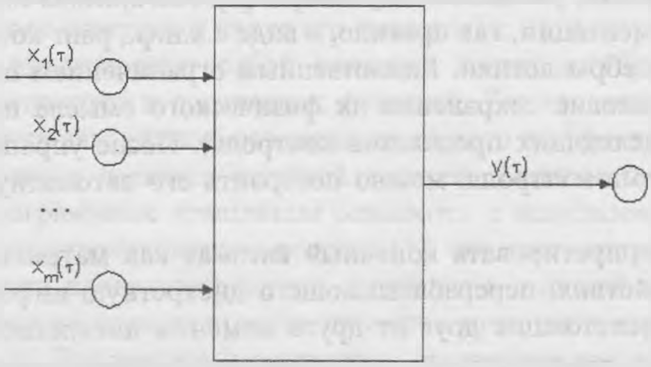
$$x_1(\tau), x_2(\tau), x_3(\tau), \ldots ,x_m(\tau) \text{ и } y(\tau).$$

Будем считать, что если $y(\tau) = 1$, то событие имело место, а при $y(\tau) = 0$ - нет. События такого рода представляют другую большую группу алгоритмов контроля. Например, событие 073 (превышение максимальной эксплуатационной скорости $V_{пр}$>575 км/ч) записывается с учетом возможной погрешности измерения скорости в виде конъюнкции предикатов следующим образом:

$$S073 = (7120 \leqslant H_б \leqslant 10300) \wedge (V_{пр} \geqslant 588). \tag{1}$$

Такое событие наступает, если хоть в одной точке полета скорость станет больше или равна 588 км/ч при высоте в указанных пределах.

Автоматы, представляющие алгоритмы контроля такого типа, так же, как и автоматы с определением события по отрезку дискретного времени, в произвольный момент времени $t=\tau$ можно представить в виде диаграммы (см. рисунок).



Функциональная схема автомата с одним выходом

Для события (1) будем иметь следующий набор функций, определенных через предикаты алгоритмов контроля:

$$x_1 = H \geqslant 7120;$$
$$x_2 = H \leqslant 10300; \tag{2}$$
$$x_3 = V \geqslant 588 .$$

Для события S096 выберем следующие функции алгебры логики, определенные на предикатах контроля:

$$x_1 = U27 \geqslant 30; \tag{3}$$
$$x_2 = \Delta\tau \leqslant 30 .$$

Причем $x_2(\tau)$ можно заменить внутренним состоянием автомата или специальным счетчиком.

Система (3) означает, что по крайней мере в 60 дискретных точках регистрации (2 точки за 1 с) значение U27 должно быть не меньше 30 В, чтобы "сработал" алгоритм контроля. События контроля S073 и S096 с учетом соотношений (2) и (3) можно записать в следующем виде:

$$S073 = (x_1 \wedge x_2) \wedge x_3 = x_1 \wedge x_2 \wedge x_3; \tag{4}$$
$$S096 = x_1 \wedge x_2. \tag{5}$$

Входной алфавит для таких событий контроля будет состоять из комбинаций 0 и 1, причем количество разрядов m в словах равно количеству функций $x_i(\tau)$, $i=1,2,\ldots,m$. После определения предиката и вычисления каждой функции в отдельности получаются наборы входных слов. Соответствующее устройство синхронной переработки информации представляет собой m-мерную булеву функцию $F(x_1(\tau),x_2(\tau),\ldots, x_m(\tau))$, заданную на множестве слов из входного алфавита. Для приведенных простых событий контроля S073 и S096 эта функция записывается с помощью соотношений (4) и (5). Функции S073 и S096 имеют элементарное табличное представление.

С помощью табличного способа задания однозначно определяется конечный структурный автомат, реализующий требуемое устройство синхронной переработки информации, а для осуществления полного цикла синтеза автомата необходимо и достаточно представить функцию, реализующую автомат в виде таблиц, и затем записать ее в форме канонических уравнений, по которым выполняется построение схемы автомата [6].

Такой подход к построению автоматов можно применить только над такими устройствами синхронной переработки информации, которые представлены ограниченно-детерминированными операторами. Детерминированность оператора $y(t) = \varphi[x(t)]$ следует из того, что при любом t буква y(t) является однозначной функцией от слова x(0), x(1), …,x(t-1), x(t) и не зависит от входных букв $x(\tau)$ при $\tau > t$. Ограниченность оператора, описывающего некоторое устройство переработки информации, означает, что среди его остаточных операторов имеется строго конечное число попарно различимых [6].

Эти условия выполняются для алгоритмов контроля, которые удовлетворяют условию детерминированности и могут быть описаны операторами без предвосхищения, условию ограниченности и являются операторами с конечной памятью, поскольку состоят из конечного числа попарно различимых предикатов.

Рассмотрим систему уравнений:

$$\begin{aligned} &z(t) = \Phi[\, x(t), q(t)\,];\\ &q(t+1) = \Psi[\, x(t), q(t)\,];\\ &t = 0,1,2,\dots\, . \end{aligned} \tag{6}$$

Выбирая произвольные функции Φ и Ψ с заданными начальными состояниями, получаем ограниченно-детерминированный оператор, вес которого не более числа букв в алфавите состояний. Уравнения (6) и функции Φ и Ψ вместе с таблицами, задающими эти функции, являются каноническими уравнениями, функциями и таблицами над алфавитами X, Z, Q (X – входной алфавит, Z – выходной алфавит, Q – алфавит состояний). В общем случае z(t), x(t), q(t), а также Φ[x(t),q(t)] и Ψ[x(t),q(t)] могут интерпретироваться как вектор. В нашей модели результирующая функция z(t) описывается одномерным вектором (поскольку на выходе имеем один полюс). Входных сигналов может быть от 1 до m, а состояние автомата q(t) представляется также одномерным вектором и зависит от вектора $x_1(t)$, $x_2(t)$, …, $x_m(t)$ и предыдущего состояния q(t-1). Поэтому система канонических уравнений (6) может быть записана в виде:

$$\begin{aligned} &z(t) = \Phi[\, x_1(t), x_2(t), \dots, x_m(t), q(t)];\\ &q(t+1) = \Psi[x_1(t), x_2(t), \dots, x_m(t), q(t)]. \end{aligned} \tag{7}$$

Уравнения (7) являются каноническими уравнениями общего вида, описывающими алгоритмы контроля ВС. Очевидно, что уравнения (7) представляют собой алгоритмы, эквивалентные исходным, так как они определены на одном и том же входном множестве и имеют однозначное отображение на одном и том же выходном множестве.

Построим, например, канонические уравнения конечного автомата, заданного ограниченно-детерминированным оператором по событию S073 (см. уравнение (4)). Для этого введем в рассмотрение состояние автомата, причем начальное состояние q(0)=0. Тогда система канонических уравнений будет иметь вид:

$$\begin{aligned} &z(t) = x_1(t) \wedge x_2(t) \wedge x_3(t);\\ &q(t+1) = x_1(t) \wedge x_2(t) \wedge x_3(t);\\ &t = 0,1,2,\dots\, . \end{aligned}$$

Состояние такого автомата в каждый момент дискретного времени зависит только от входных сигналов, поступивших в данный момент, и не зависит от состояния автомата в предыдущие моменты времени. Поэтому автоматы, реализующие алгоритмы контроля такого вида, функционируют как обычные переключатели.

Канонические уравнения автомата, заданного оператором по событию S096 (см. уравнение (5)) будут иметь следующий вид:

$$\begin{aligned} &z(t) = x_1(t) \wedge (q(t) \geqslant 30);\\ &q(t+1) = (x_1(t) + q(t)) \wedge x_1(t);\\ &t = 0,1,2,\dots\, . \end{aligned} \tag{8}$$

Для вычисления суммы в уравнении состояния применим сумматор последовательного действия. Тогда система запишется в виде двух систем, определяющих композицию структурных автоматов:

$$
\begin{aligned}
&y(t) = x_1(t) \oplus q(t) \oplus q_\Sigma(t);\\
&_\Sigma(t+1) = x_1(t)q(t) \vee x_1(t)q_\Sigma(t) \vee q(t)q_\Sigma(t);\\
&q_\Sigma(0) = 0;\\
&t = 0,1,2,\ldots .
\end{aligned}
\tag{9}
$$

$$
\begin{aligned}
&z(t) = x_1(t) \wedge (q(t) \geqslant 30);\\
&q(t+1) = y(t)) \wedge x_1(t);\\
&q(0) = 0;\\
&t = 0,1,2,\ldots .
\end{aligned}
$$

В дальнейшем при построении логической схемы автомата необходимо учитывать, что система (9) описывает суммирование в общем виде, без учета разрядности сумматора, и поэтому она может быть раскрыта, например, в 16 систем, каждая из которых будет описывать суммирование в соответствующем разряде шестнадцатиразрядного сумматора последовательного действия.

Используя приведенную методику построения канонических уравнений конечных автоматов как для одномоментного алгоритма, так и для алгоритма по отрезку дискретного времени, можно формализованно описать любой алгоритм контроля ВС.

Полученная автоматная модель описывает устройство синхронной переработки информации, реализующее тот или иной алгоритм контроля полетов. Такая модель может быть эффективно представлена в виде программы с использованием Р-технологии программирования [7], которая имеет существенно автоматную природу, поскольку базируется на работах академика В.М.Глушкова по конечным автоматам и автоматам с магазинной памятью.

Список литературы

1. *Малежик А.И.* Основы компьютерных технологий оперативного контроля полетов воздушных судов по полетной информации. –К: КМУГА, 1996. – 124 с.

2. *Яцков Н.А.*. Основы построения автоматизированных систем контроля полетов воздушных судов. –К: КИИГА, 1989. – 344 с.

3. *Райчев И.Э.,Харченко А..Г., Яцков Н.А.* Исследование методов тестирования программных модулей обработки полетной информации// Вестн. КМУГА. – 2000. –№ 1-2. – С. 127–133.

4. *Райчев И.Э., Харченко А.Г., Яцков Н.А.* Методы создания тестовых наборов данных при сертификационных испытаниях комплексов программ контроля полетов// Вісн. НАУ. –2001. – № 1. – С. 126–132.

5. *Глушков В.М.*Синтез цифровых автоматов. –М.: Физматгиз, 1962. – 476 с.

6. *Кобринский Н.Е., Трахтенброт Б.А.* Введение в теорию конечных автоматов. –М.: Физматгиз, 1962. – 404 с.

7. *Вельбицкий И.В.* Технология программирования. – К.: Техніка, 1984. – 279 с.

Стаття надійшла до редакції 06.07.01.