

Запропонована новітня процедура вичленення випадкових коефіцієнтів в автономну групу з матричного оператора досліджуваної динамічної системи підвищує точність оцінки показників якості при стохастичних впливах.

Список літератури

1. Матвеев П.С., Синицын А.С. Динамическая точность систем автоматического управления со случайными параметрами // Автоматическое управление и вычислительная техника. – 1964, – № 6. – С. 72–84.
2. Блохин Л.Н., Кадышев И.К., Трифонов-Богданов П.И. Основы навигации и пилотажно-навигационные комплексы: Учеб. для вузов ГА / Под ред. Л.Н. Блохина. – М.: Воздуш. трансп. 1993. – 244 с.

Стаття надійшла до редакції 10.04.02.

УДК 629.735.05:681.178:519.713 (045)

І.Е. Райчев, асист.

СИНТЕЗ АВТОМАТНИХ МОДЕЛЕЙ КОНТРОЛЮ

Розглянуто питання застосування скінченних автоматів для оброблення інформації параметричних бортових реєстраторів. Наведено аналіз етапів синтезу скінченних автоматів, які зображають алгоритми контролю польотів повітряних суден.

Вступ. Інформація параметричних бортових реєстраторів використовується для вирішення задач контролю режимів польоту і перевірки виконання правил льотної експлуатації повітряних суден (ПС), оцінки працездатності авіаційної техніки й визначення причин авіаційних подій і інцидентів. Польотна інформація (ПІ), записана бортовими засобами реєстрації, обробляється згідно з алгоритмами контролю на борту ПС чи наземними системами після його посадки [1; 2; 3]. Алгоритми контролю, що складаються розробником ПС, являють собою складні і громіздкі логіко-алгебричні вирази, які містять десятки предикатів й логічних функцій з граничними умовами і регламентують поведінку ПС як об'єкта контролю протягом усіх етапів польоту. Наприклад, для літака Ту-154Б задано 172 події контролю пілотування і 118 подій контролю працездатності, причому до більшості логічних визначень включаються умови готовності й ознаки етапів.

Оскільки вхідна інформація для алгоритмів контролю має вигляд дискретної скінченної множини значень, а вихідна є скінченною множиною дискретних подій, то для реалізації таких алгоритмів природно буде застосувати математичний апарат дискретних скінченних автоматів [4]. Виходячи з канонічних рівнянь скінченного автомата і його логічної схеми, можна побудувати відповідну цьому автоматів формальну граматику [5]. За допомогою таких граматики можна описати будь-який набір алгоритмів контролю. Використовуючи, наприклад, граматику на графах – Р-граматику, що має істотно автоматну природу, можна ефективно реалізувати алгоритми контролю різних типів ПС у вигляді програмних модулів, що визначають події контролю і побудованих відповідно до Р-технології програмування [6; 7].

Розглянуті питання синтезу логічних і електричних схем автоматних моделей надалі можна використовувати для побудови логічних і апаратних схем відповідних алгоритмів контролю. Такі автоматні моделі можуть бути застосовані для контролю об'єктів, алгоритми контролю яких мають вигляд логічних функцій з обмеженнями на контрольовані параметри протягом інтервалу часу (динамічні об'єкти, стаціонарні об'єкти, стан яких оцінюється за допомогою вимірювальних приладів чи датчиків).

Класифікація алгоритмів контролю. Алгоритми контролю пілотування та працездатності ПС описують ситуації контролю і являють собою в загальному випадку суперпозицію логічних функцій, кожна з яких складається зі скінченного числа предикатів. Базовими операціями в цих функціях алгебри логіки є кон'юнкція, диз'юнкція та заперечення. Алгоритми контролю

ПС подані в технічній документації здебільшого у вигляді досконалої кон'юнктивної нормальної форми і поділяються на два класи:

- 1) алгоритми контролю без тривалості в часі;
- 2) алгоритми контролю з тривалістю в часі.

Алгоритм контролю, що належить першому класу, «спрацьовує» у деякій точці дискретного (автоматного) часу, якщо логічна функція, що визначає даний алгоритм, дорівнює 1 у цій точці і «не спрацьовує», якщо ця функція дорівнює 0. Автомати, що реалізують алгоритми контролю такого виду, є перемикачами. Сукупність предикатів, присутніх у подіях контролю, утворюють базову множину алгоритмів контролю першого класу. Вони мають вид рівностей чи нерівностей, наприклад, $V_{пр} \geq 410$, $n_{нд1} \geq 28,0$, $\alpha_{руд} < 110$, $N_r \geq 15$ та ін.

Другий клас алгоритмів контролю описує події, настання яких залежить від величини відрізка дискретного часу, на якому логічна функція алгоритму приймає значення 1 (тобто враховується «історія» події). Якщо тривалість такого відрізка часу перевищує якусь наперед задану величину, то подія контролю вважається такою, що відбулася. Основну групу алгоритмів контролю другого класу складають ознаки етапів і готовності подій, що присутні як уточнюючі події в більшості алгоритмів контролю. Один раз побудувавши автомат, що описує ознаку етапу чи готовність, можна багаторазово використовувати його копію в інших алгоритмах контролю.

Алгоритми контролю першого класу в більшості випадків можна задати за допомогою абстрактних автоматів Мілі чи Мура. Для опису більш складних алгоритмів контролю другого класу потрібно використати апарат структурних автоматів і логічних схем [8; 9]. Задасмося метою визначити загальні підходи побудови канонічних рівнянь і логічних схем автоматів за алгоритмами контролю, тобто тих компонентів, що однозначно визначають скінченний автомат і необхідні для його синтезу.

Загальні правила побудови канонічних рівнянь скінченних автоматів за алгоритмами контролю. Канонічні рівняння задають функцію автомата, а також функцію його стану в часі. Як впливає з визначення, стан автомата, що реалізує алгоритм контролю першого класу, збігається з його вихідним значенням. Тому при написанні канонічних рівнянь варто враховувати, що рівняння стану повинне збігатися за правою частиною з рівнянням функції автомата. Нехай, наприклад, алгоритм контролю записується у вигляді

$$(x_1 \vee x_2) \cdot x_3 \cdot x_4,$$

де x_1, x_2, x_3, x_4 – суть предикати контролю.

Тоді система канонічних рівнянь буде мати вигляд:

$$z(t) = (x_1 \vee x_2) \cdot x_3 \cdot x_4;$$

$$q(t+1) = (x_1 \vee x_2) \cdot x_3 \cdot x_4; \quad (1)$$

$$q(0) = 0;$$

$$t = 0, 1, 2, \dots,$$

де $z(t)$ – функція автомата; $q(t)$ – його стан.

Тут $z(t)$ і $q(t)$ приймають тільки два значення: 0 чи 1.

Для алгоритмів контролю другого класу побудувати канонічні рівняння виявляється значно складніше. Для того, щоб врахувати тривалість події контролю, внесемо її величину в стан автомата. При цьому для накопичення суми необхідно забезпечити порозрядне додавання по mod2 із залученням елементів затримки на 1 такт. Один з можливих підходів при реалізації цього методу проілюструємо на прикладі.

Нехай функція автомата має вигляд:

$$x_1 \cdot x_2 \cdot x_3 \cdot (x_4 \vee x_5),$$

а необхідна тривалість дорівнює 600 точкам реєстрації. Позначимо

$$z_0(t) = x_1(t) \cdot x_2(t) \cdot x_3(t) \cdot (x_4(t) \vee x_5(t)). \quad (2)$$

Тоді система канонічних рівнянь буде мати вигляд:

$$\begin{aligned} z(t) &= z_0(t) \cdot (q(t) \geq 600); \\ q(t+1) &= z_0(t) \cdot y(t); \\ q(0) &= 0; \\ t &= 0, 1, 2, \dots, \end{aligned} \quad (3)$$

де $y(t)$ є двійковим суматором послідовної дії з використанням елементів затримки на один такт, що, у свою чергу, записується за допомогою двох систем канонічних рівнянь:

$$\begin{aligned} y(t) &= z_0(t) \oplus y_z(t) \oplus q_\Sigma(t); \\ q_\Sigma(t+1) &= z_0(t) \cdot y_z(t) \vee z_0(t) \cdot q_\Sigma(t) \vee y_z(t) \cdot q_\Sigma(t); \end{aligned} \quad (4)$$

$$\begin{aligned} q_\Sigma(0) &= 0; \\ y_z(t) &= q_z(t); \\ q_z(t+1) &= y(t); \\ q_z(0) &= 0. \end{aligned} \quad (5)$$

Визначивши систему канонічних рівнянь (3) з додатковими системами (4) і (5), ми побудували одну з можливих моделей, що описують скінченний структурний автомат, функція якого полягає в пошуку події контролю [4]. Подібним способом будують будь-які інші автомати, що реалізують алгоритми контролю другого класу. Трохи іншу модель можна одержати, якщо величину тривалості записувати в спеціальний лічильник, хоча і при такому підході необхідно використовувати послідовне порозрядне додавання.

Системи (4) і (5) розпадаються до 16 однотипних систем кожна, відповідно до розрядності суматора. Для забезпечення переносу з молодших розрядів у старші система (4) повинна бути спрощена [8; 10]. Запропонований математичний апарат дозволяє будувати канонічні рівняння скінченних автоматів для будь-яких алгоритмів контролю першого і другого класу.

Правила побудови логічних схем алгоритмів контролю. Логічні схеми будуються на основі канонічних рівнянь і відображають логіку функціонування автомата, включаючи логічні операції і шляхи проходження інформації починаючи від вхідних і закінчуючи вихідними полюсами [9].

Для алгоритмів контролю першого класу, які мають вигляд простих скінченних автоматів (перемикачів), логічні схеми будуються елементарним чином. Схеми таких автоматів містять у собі тільки функцію автомата. Побудуємо, наприклад, логічну схему скінченного автомата за канонічними рівняннями (1). Функція цього автомата записується так:

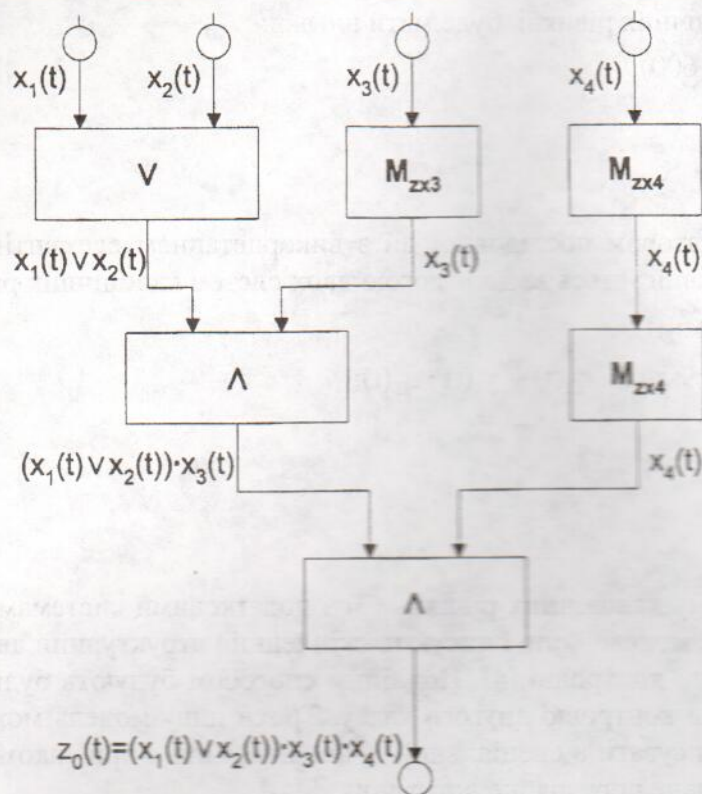
$$z(t) = (x_1(t) \vee x_2(t)) \cdot x_3(t) \cdot x_4(t).$$

Відповідну логічну схему автомата зображено на рис. 1, а, де використані логічні операції \wedge і \vee , а також елементи затримки для предикатів $x_3(t)$ і $x_4(t)$.

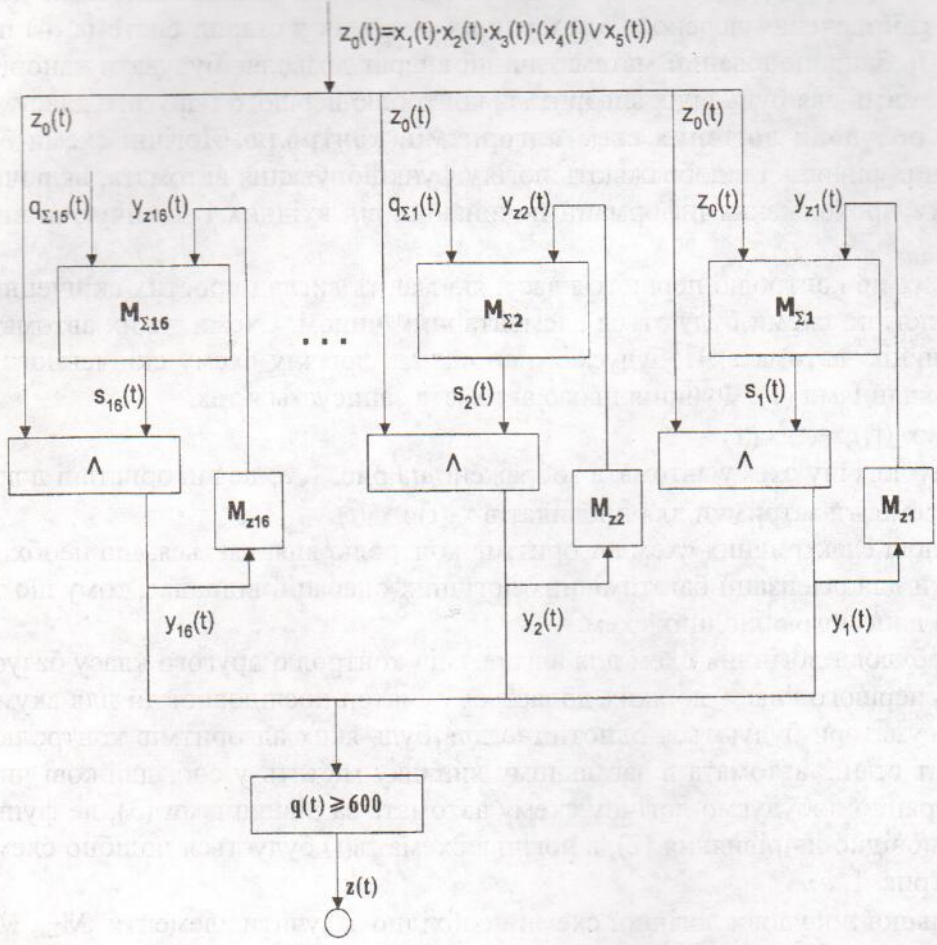
При побудові електричних схем алгоритмів контролю виявляється, що необхідність в елементах затримки для реалізації багатомісних логічних операцій відпадає, тому що такі операції реалізуються за допомогою діодних схем.

Процес побудови логічних схем для алгоритмів контролю другого класу базується на схемах алгоритмів першого класу, до яких додається суматор послідовної дії для акумуляції стану автомата. Такі суматори будуються однотипно для будь-яких алгоритмів контролю 2-го класу, оскільки алфавіт станів автомата в загальному випадку містить у собі двійкові числа від 0 до $2^{15}-1$. Як ілюстрацію побудуємо логічну схему автомата за рівняннями (3), де функція $z_0(t)$ обчислюється відповідно до рівняння (2), а логічна схема $z_0(t)$ будується подібно схемі алгоритму першого класу (рис. 1, а).

Для подальшої побудови логічної схеми необхідно залучити елементи $M_{\Sigma 1}, M_{\Sigma 2}, \dots, M_{\Sigma 16}$, які являють собою суматори послідовної дії для кожного розряду стану $y(t)$ і переносів $q_\Sigma(t)$. Канонічні рівняння таких суматорів записуються відповідно до модифікованої системи (4), а



a



b

Рис. 1. Логічна схема автомата за алгоритмом контролю першого (а) та другого (б) класу

рівняння елементів затримки $M_{Z1}, M_{Z2}, \dots, M_{Z16}$ – відповідно до системи (5). Подальшу частину логічної схеми показано на рис. 1, б.

У такий спосіб (за допомогою композиції двох схем) можна одержати логічну схему автомата для будь-якого алгоритму контролю другого класу. Порівняння вигляду $q(t) \geq 600$ і предикати $V_{пр} \geq 410$, $H_r < 15$ реалізуються перемикачами.

Приклад синтезу скінченного автомата за алгоритмом контролю. Розглянемо алгоритм контролю S068 контролю пілотування для літака Ту-154Б (ресстратор МСРП-64-2): «Превышение скорости полета при уборке шасси»:

$$S068 = \overline{ГТ_{у.ш}} \wedge \overline{i_{ш.в}} \wedge (V_{пр} \geq 410),$$

де «готовність уборки шасси» $ГТ_{у.ш}$ виставляється в одиницю у випадку $\overline{i_{ш.в}} \wedge (\Delta\tau \geq 3)$, тобто шасі прибрано протягом 3 с і більше. Тоді подію S068 можна зобразити у вигляді:

$$S068 = (\overline{i_{ш.в}} \wedge (\Delta\tau \geq 3)) \wedge \overline{i_{ш.в}} \wedge (V_{пр} \geq 410),$$

де $ГТ_{у.ш}$ є алгоритмом контролю другого класу, а в цілому S068 можна зобразити як композицію заперечення готовності прибирання шасі й алгоритму контролю першого класу, що складається з кон'юнкції двох предикатів $\overline{i_{ш.в}}$ і $(V_{пр} \geq 410)$. Тому для події S068 маємо такий набір елементарних предикатів:

$$x_1 = \overline{i_{ш.в}}; \quad \overline{x_1} = i_{ш.в}; \quad x_2 = V_{пр} \geq 410.$$

Функцію тривалості $\Delta\tau \geq 3$ реалізуємо за допомогою спеціального суматора, організацію якого розглянемо трохи пізніше. Тоді подія S068 запишеться у вигляді:

$$S068 = \overline{x_1} \wedge (\Delta\tau \geq 3) \wedge \overline{x_1} \wedge x_2.$$

Вихідна функція автомата за алгоритмом контролю другого класу $z_0(t)$ (без урахування тривалості) для кожного моменту дискретного часу має вигляд:

$$z_0(t) = \overline{x_1}(t).$$

Стан автомата $q(t)$ буде містити в собі тривалість інтервалу, протягом якого відбувалася подія $z(t)$. Для реалізації такого підходу використаємо такі елементарні автомати: M_z – елемент затримки на один такт і M_Σ – суматор послідовної дії.

Оскільки тривалості інтервалів контролю для літака Ту-154 можуть перевищувати 615 с, відповідний стан буде містити число більше чи рівне 1230 підряд розташованих точок польоту. У загальному випадку для зображення такого стану буде потрібно слово, що складається не більш, ніж з 16 розрядів. Враховуючи це, система канонічних рівнянь для $ГТ_{у.ш}$ набуває такого вигляду:

$$z(t) = z_0(t) \cdot (q(t) \geq 3);$$

$$q(t+1) = z_0(t) \cdot y(t);$$

$$q(0) = 0.$$

де $y(t)$ являє собою суматор послідовної дії з елементами затримки на один такт. Він може бути описаний системами рівнянь (4) і (5), причому система (4) описує суматор послідовної дії M_Σ , а система (5) – елемент затримки M_z .

Побудувавши систему канонічних рівнянь (6) з додатковими системами (4), (5), ми визначили скінченний автомат, функція якого полягає у відстеженні події «готовність уборки шасси» за копією польоту. Цей автомат може використовуватися для автоматизованого пошуку контрольованої події за ПІ. До складу автомата, визначеного системою рівнянь (6), входить ряд елементарних автоматів (4), (5). Повну логічну схему алгоритму контролю S068 побудуємо, ґрунтуючись на системах (4), (5), (6).

Внутрішній стан суматора (вектор переносів) $q_\Sigma(t)$ складається з 15 розрядів: $q_\Sigma(t) = q_{15}q_{14} \dots q_2q_1$. Можна виробляти також і 16-й розряд q_{16} , однак будемо вважати, що $q_{16} = 0$, оскільки, по-перше, тривалості в подіях контролю для даного типу ПС не перевищують 15 хв,

Отже, розроблена логічна схема автомата (рис. 2), що функціонує відповідно до алгоритмів контролю S068. Ця схема спирається на канонічні рівняння (5)–(10).

Наступним кроком є побудова електричної схеми скінченного автомата за алгоритмом S068, що, у свою чергу, базується на логічній схемі, яка показана на рис. 2. Будемо користатися тим фактом, що в електричній схемі можна реалізувати багатомісні операції кон'юнкції і диз'юнкції без використання елементів затримки. Крім того, при побудові електричної схеми будемо використовувати оператор тріода $T(x) = \bar{x}$, що реалізує функцію заперечення. Для реалізації елемента затримки на один такт скористаємося оператором $z(t+1) = G(x,0) = x(t)$, що будуватиметься за допомогою феромагнітного елемента. Двійковий суматор будемо організовувати на базі систем рівнянь (8), причому для побудови електричної схеми автомата зручно буде скористатися системами, що утворюються із системи (8) за допомогою елементарного еквівалентного перетворення операції додавання за модулем 2:

$$\begin{aligned}
 y_1(t) &= z_0(t) \overline{y_{z1}}(t) \vee \overline{z_0}(t) y_{z1}(t); \\
 q_{\Sigma 1p}(t) &= z_0(t) y_{z1}(t); \\
 q_{\Sigma 1p}(0) &= 0; \\
 y_2(t) &= q_{\Sigma 1p}(t) \overline{y_{z2}}(t) \vee \overline{q_{\Sigma 1p}}(t) y_{z2}(t); \\
 q_{\Sigma 2p}(t) &= q_{\Sigma 1p}(t) y_{z2}(t); \\
 q_{\Sigma 2p}(0) &= 0; \\
 &\dots\dots\dots \\
 y_{15}(t) &= q_{\Sigma 14p}(t) \overline{y_{z15}}(t) \vee \overline{q_{\Sigma 14p}}(t) y_{z15}(t); \\
 q_{\Sigma 15p}(t) &= q_{\Sigma 14p}(t) y_{z15}(t); \\
 q_{\Sigma 15p}(0) &= 0; \\
 y_{16}(t) &= q_{\Sigma 15p}(t) \overline{y_{z16}}(t) \vee \overline{q_{\Sigma 15p}}(t) y_{z16}(t); \\
 q_{\Sigma 16p}(t) &= 0; \\
 q_{\Sigma 16p}(0) &= 0.
 \end{aligned} \tag{11}$$

На базі систем (6), (9), (10), (11) можна побудувати принципову електричну схему скінченного автомата, що наведена на рис. 3. Порівняння $q(t) \geq 3$ реалізується за допомогою елементарної чотирирозрядної діодної схеми. При побудові логічної і принципіальної електричної схем алгоритмів другого класу суматори будуються однотипним чином і можуть містити стільки розрядів, скільки необхідно для накопичення тривалості події (наприклад, для тривалості 3 – чотири розряди, для тривалості 30 – шість розрядів та ін.), причому максимально не більше 16 розрядів.

Висновок. Скінченний структурний автомат, що описує алгоритм контролю першого чи другого класу, синтезований, якщо побудовані такі його компоненти: канонічні рівняння, логічна схема, принципова електрична схема.

При побудові електричної схеми використовуються напівпровідникові елементи, що одержали широке поширення. На базі цих елементів можна побудувати адаптер для комп'ютера, що буде «відслідковувати» за копією польоту настання довільної (досить великої) множини подій контролю за допомогою інтегрованих на ньому розпізнавальних автоматів. Реалізувати таку множину алгоритмів контролю можна також і за допомогою спеціально запрограмованої мікросхеми з розпізнавальними автоматами [12]. При цьому отримаємо економічно ефективні автоматні моделі з найменшою можливою кількістю елементів. Отже, за допомогою апаратних засобів побудована процедура пошуку подій контролю. Якщо додати до цієї процедури методіку побудови Р-програм за алгоритмами контролю, то одержимо повний цикл виявлення подій контролю як апаратними, так і програмними засобами [2].

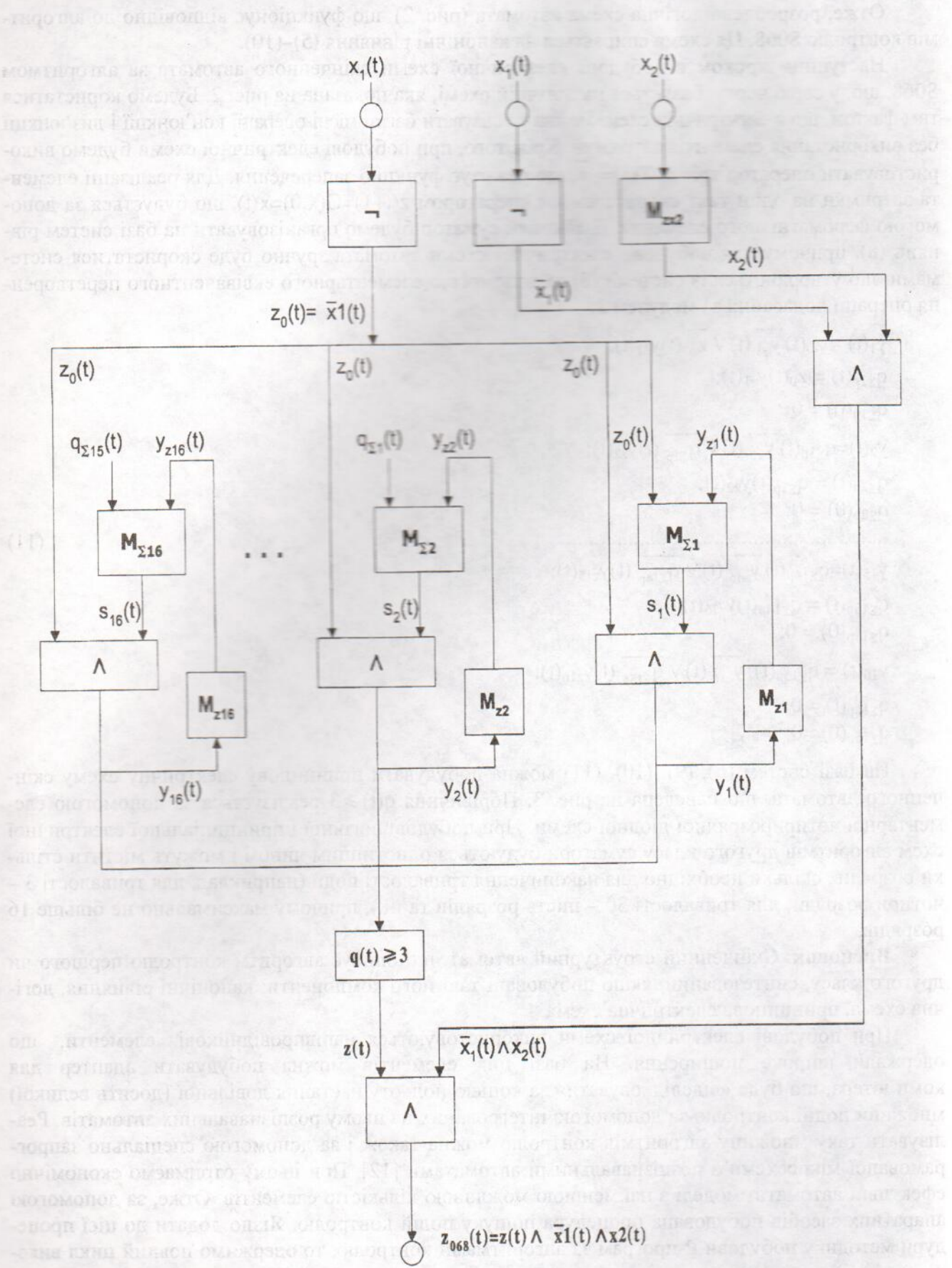


Рис. 2. Повна логічна схема алгоритму контролю S068

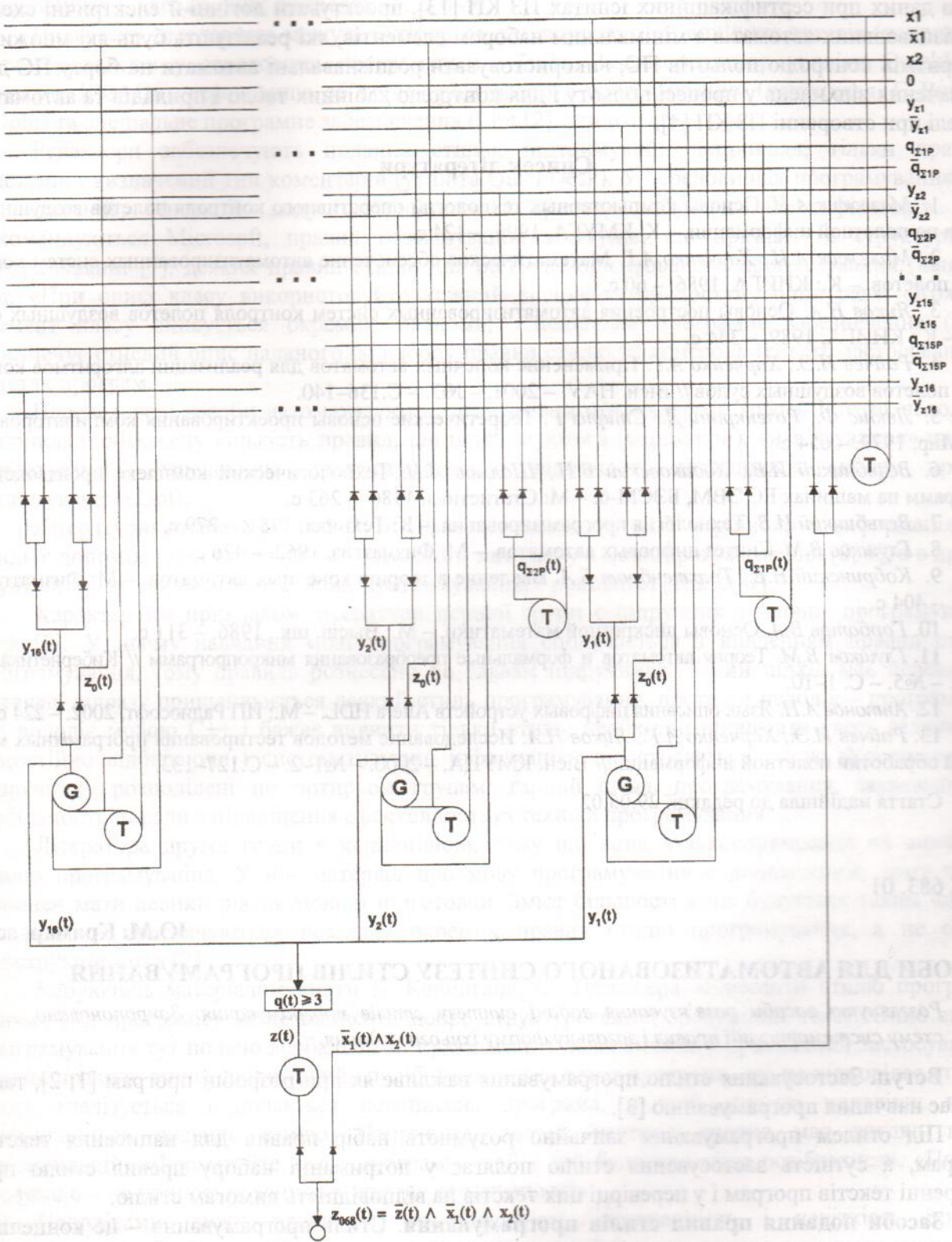


Рис. 3. Принципова електрична схема автомата за подією контролю S068

Застосування автоматних моделей контролю дає можливість підвищити ефективність тестування компонент програмного забезпечення контролю польотів (ПЗ КП) шляхом автоматизації пошуку моментів контролю за ПП, здійснити генерацію і внесення подій контролю в копію польоту, автоматизувати пошук етапів і моментів контролю в системах створення тестових наборів даних при сертифікаційних іспитах ПЗ КП [13], проектувати логічні й електричні схеми розпізнавальних автоматів з мінімальним набором елементів, які реалізують будь-які множини алгоритмів контролю польотів ПС, використовувати розпізнавальні автомати на борту ПС для визначення відхилень у процесі польоту і для контролю кабіних панелей і приладів та автоматні моделі при створенні ПЗ КП [4].

Список літератури

1. *Малежик А.И.* Основы компьютерных технологий оперативного контроля полетов воздушных судов по полетной информации. – К: КМУГА, 1996. – 124 с.
2. *Малежик А.И., Харченко А.Г.* Математическое обеспечение автоматизированных систем контроля полетов. – К.: КИИГА, 1986. – 60 с.
3. *Яцков Н.А.* Основы построения автоматизированных систем контроля полетов воздушных судов. – К.: КИИГА, 1989. – 344 с.
4. *Райчев И.Э., Харченко А.Г.* Применение конечных автоматов для реализации алгоритмов контроля полетов воздушных судов // Вісн. НАУ. – 2001. – №3. – С.136–140.
5. *Льюис Ф., Розенкранц Д., Стирнз Р.* Теоретические основы проектирования компиляторов. – М.: Мир, 1979. – 654 с.
6. *Вельбицкий И.В., Ходаковский В.Н., Шолмов Л.И.* Технологический комплекс производства программ на машинах ЕС ЭВМ, БЭСМ-6. – М: Статистика, 1980. – 263 с.
7. *Вельбицкий И.В.* Технология программирования. – К: Техника, 1984. – 279 с.
8. *Глушков В.М.* Синтез цифровых автоматов. – М: Физматгиз, 1962. – 476 с.
9. *Кобринский Н.Е., Трахтенброт Б.А.* Введение в теорию конечных автоматов. – М: Физматгиз, 1962. – 404 с.
10. *Горбатов В.А.* Основы дискретной математики. – М.: Высш. шк., 1986. – 311 с.
11. *Глушков В.М.* Теория автоматов и формальные преобразования микропрограмм // Кибернетика. – 1965. – №5. – С. 1–10.
12. *Антонов А.Н.* Язык описания цифровых устройств Altera HDL. – М: ИП Радиософт, 2002. – 224 с.
13. *Райчев И.Э., Харченко А.Г., Яцков Н.А.* Исследование методов тестирования программных модулей обработки полетной информации // Вісн. КМУЦА. – 2000. – №1–2. – С.127–133.

Стаття надійшла до редакції 29.05.02.

УДК 683.01

Ю.М. Крамар, асп.

ЗАСОБИ ДЛЯ АВТОМАТИЗОВАНОГО СИНТЕЗУ СТИЛІВ ПРОГРАМУВАННЯ

Розглянуто засоби розв'язування задачі синтезу стилів програмування. Запропоновано схему систематизації правил і загальну форму їхнього подання.

Вступ. Застосування стилю програмування важливе як при розробці програм [1; 2], так і під час навчання програмуванню [3].

Під стилем програмування звичайно розуміють набір правил для написання текстів програм, а сутність застосування стилю полягає у дотриманні набору правил стилю при створенні текстів програм і у перевірці цих текстів на відповідність вимогам стилю.

Засоби подання правил стилів програмування. Стиль програмування – це концепція написання програм, що пронизує всі процеси створення програмного забезпечення та подається за допомогою набору правил. Стиль відбиває не тільки технічний, але й культурний досвід, який існує у період життєвого циклу програмного забезпечення [4]. Беручи до уваги