УДК 681.5

І.О. ФУРМАН, С.Я. БОВЧАЛЮК

Харківський національний технічний університет сільського господарства ім. Петра Василенка, Україна

МАТЕМАТИЧНА МОДЕЛЬ ПАРАЛЕЛЬНОГО КЕРУЮЧОГО АВТОМАТА ПІДВИЩЕНОЇ НАДІЙНОСТІ

Наведено принцип функціонування паралельного логічного контролера. Представлена математична модель паралельного керуючого автомата підвищеної надійності та розширеними функціональними можливостями.

програмований логічний контролер, паралельна архітектура, математична модель, надійність

Вступ

Аналіз стану питання. Одними з перших робіт по функціональним моделям паралельних автоматів були фундаментальні дослідження О. Д. Закревського [1]. Однак практичної реалізації керуючих автоматів на основі цих моделей не було. У роботі [2] запропонована математична модель логічного керуючого автомата паралельної дії, на базі якої було розроблено архітектуру, реалізовано та впроваджено у виробництво функціонально завершені програмовані логічні контролери (ПЛК) паралельної дії (паралельні ПЛК).

Аналіз моделі, що описана в [2], та досвід практичного використання паралельних ПЛК показав, що вказана модель має суттєві недоліки: В ній не врахована можливість автоматичного виявлення та заборони видачі аварійних комбінацій вихідних сигналів; крім того, у моделі обмежені можливості логічного аналізу потоку станів входів.

Метою статті є розробка математичної моделі логічного керуючого автомата паралельної дії в якій усунені недоліки базової моделі, та розширені функціональні можливості автомата, що будується на її основі.

Основні матеріали дослідження

При побудові базової моделі використана наступна концепція уявлення послідовних та паралельних ПЛК, в основі якої лежить класифікація їх за принципом взаємодії контролера з керованим об'єктом. Як послідовні, так і паралельні ПЛК в загальному випадку можна характеризувати тривалістю циклу однократного обслуговування контрольованих входів і керованих виходів

$$T_ц = t^{\mathrm{I}} + t^{\mathrm{II}},$$

де $T_ц$– тривалість циклу однократного обслуговування всіх входів-виходів контролера в одиницях дискретного автоматного часу; t^{I} – дискретний автоматний час, що витрачається на аналіз станів входів; t^{II} – дискретний автоматний час, що витрачається на формування команд керування.

До послідовних ПЛК відносяться контролери, в яких використовується послідовний принцип обслуговування контрольованих входів і керованих виходів. Для таких контролерів маємо:

$$t^{\mathrm{I}} = \sum_{i=1}^{k} t_i^{\mathrm{I}}\ ;\ \ t^{\mathrm{II}} = \sum_{j=1}^{m} t_j^{\mathrm{II}}\ ,$$

де k – кількість контрольованих входів; m – кількість команд керування, що формуються.

Паралельними ПЛК називаються такі контролери, в яких використовується паралельний принцип обслуговування входів-виходів і для яких:

$$t^{\mathrm{I}} = t_i^{\mathrm{I}} (i = 1, 2, ..., k)\ ;\ \ t^{\mathrm{II}} = t_j^{\mathrm{II}}\ (j = 1, 2, ..., m)\ .$$

В основу побудови математичної моделі функціонування паралельного ПЛК, покладено узагальнене формалізоване описання поведінки технологічних агрегатів дискретної циклічної дії ($\text{ТА}_{\text{ДЦ}}$).

$\text{ТА}_{\text{ДЦ}}$ – це об'єкти, поведінка яких в часі та просторі строго детермінована [2] і може бути описане кінцевою кількістю станів, в яких протягом циклу роботи агрегату знаходяться механізми:

$$C = \{c_1, c_2, \ldots, c_m\}$$

і кінцевою кількістю станів датчиків, що контролюють стани відповідних механізмів:

$$A = \{a_1, a_2, \ldots, a_k\}.$$

Цикл роботи $\text{ТА}_{\text{ДЦ}}$ являє собою кінцеву кількість інтервалів дискретного автоматного часу:

$$T = \{t_1, t_2, \ldots, t_s\},$$

причому для кожного i-го інтервалу існує кінцева підмножина (комбінація) станів датчиків:

$$A_i = \{a_{i1}, a_{i2}, \ldots, a_{ik}\},\ A_i \subset A,$$

яка є єдиною "дозволеною" для включення відповідної єдиної кінцевої підмножини (комбінації) механізмів:

$$C_i = \{c_{i1}, c_{i2}, \ldots, c_{im}\},\ C_i \subset C.$$

Отже, циклограма роботи $\text{ТА}_{\text{ДЦ}}$ може бути задана двома прямокутними матрицями кінцевих розмірів: матрицею станів датчиків A і матрицею станів механізмів C із детермінованим розміщенням у них векторів-рядків, причому кожному i-му рядку матриці A однозначно відповідає i-й рядок матриці C:

$$A = \begin{Vmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \cdots & \cdots & \cdots & \cdots \\ a_{s1} & a_{s2} & \cdots & a_{sk} \end{Vmatrix};\quad C = \begin{Vmatrix} c_{11} & c_{12} & \cdots & c_{1m} \\ c_{21} & c_{22} & \cdots & c_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ c_{s1} & c_{s2} & \cdots & c_{sm} \end{Vmatrix}.$$

Кількість стовпців k матриці A відповідає кількості контрольованих датчиків, встановлених на механізмах технологічного агрегату, а кількість стовпців m в матриці C – числу керованих механізмів агрегату. В загальному випадку $k \neq m$. Кількість рядків s в обох матрицях однакова і дорівнює числу рядків (кроків, етапів) циклограми.

Оскільки управління технологічним устаткуванням дискретної дії здійснюється за звичай за кінцевими станами механізмів і датчиків ("включено" – "виключено"), то циклограма роботи такого устаткування може бути описана за допомогою булевих матриць. Таким чином, як математичну модель функціонування $\text{ТА}_{\text{ДЦ}}$ можна використати булеві матриці типу C_{im} і A_{ik} з детермінованим розміщенням в них векторів-рядків станів механізмів і векторів-рядків станів відповідних датчиків.

Для формального описання функціонування ПЛК, як керуючого автомата, необхідно розглянути взаємодію наступних векторів:

– вектор a – станів (комбінація станів органів керування та датчиків керованого ТА, значення яких можуть бути чітко визначені в часі, тобто є детермінованими);

– вектор $в$ – умов (комбінація станів зовнішнього середовища, в якості якого можуть виступати як органи керування і датчики, що встановлені на керованому технологічному агрегаті, так і сигнали суміжних систем керування, які в загальному випадку є умовами розгалуження алгоритму керування і заздалегідь не можуть бути визначені, тобто є стохастичними);

– вектор c – вектор керування (комбінація сигналів, що подаються на виконавчі механізми технологічного агрегату).

Визначимо також додаткові вектори, необхідні для опису внутрішніх станів керуючого автомата:

– вектор d – адреси переходу (являє собою номер рядка матриць A і C (тобто адресу) початку підпрограми в алгоритмі з розгалуженнями);

– вектор e – заборонених станів (комбінація команд керування виконавчими механізмами, яка є забороненою і поява якої на виході керуючого автомата може призвести до аварійного стану керованого об’єкта).

Множина векторів a_i утворює матрицю A – станів, розміром $s \times k$, а множина векторів c_i формує матрицю C – команд, розміром $s \times m$, призначення яких було надано вище. При цьому кожному з векторів a і c поточних значень станів і керувань для кожного моменту дискретного автоматного часу ставиться у відповідність певний рядок матриць A і C.

Матриця B – адрес переходів розміром $q \times u$. Складається з рядків, кожний з яких є умовою переходу до підпрограми і кількість яких визначається алгоритмом функціонування технологічного агрегату. Кількість стовпців матриці визначається розміром вектора $в$. Необхідно зазначити, що в процесі роботи керуючого автомата відбувається відображення рядка матриці B у вектор d, при цьому розміри векторів $в$ і d у загальному випадку не співпадають, тобто $u \neq \log_2(s)$.

Для покращення показників надійності роботи керованого об’єкта і для унеможливлення видачі керуючим автоматом комбінації вихідних сигналів, що можуть призвести до аварії, використовується матриця E – заборонених станів. Число рядків і стовпців матриці визначається алгоритмом функціонування керованого об’єкта і в загальному випадку дорівнює $r \times m$, де m – кількість керуючих сигналів, а r – кількість заборонених їх комбінацій.

Розглянемо тепер алгоритм взаємодії вказаних векторів, який буде визначати принцип функціонування логічного керуючого автомата паралельної дії (ЛКАП). Хай на p-му кроці алгоритму функціонування керованого об’єкта в проміжку між моментами часу t_p і t_{p+1} вектор керувань $C(t_p)$ отримує значення компонент $c_j(t_p)$, що списані з i-го рядка матриці програми керування C і зберігає їх до початку наступного, p+1-го кроку алгоритму

$$c_j(t_p) = C_{ij},$$

де індекс j пробігає значення від 1 до m, а номер рядка визначається на $(p-1)$-му кроці алгоритму

$$I = g(p-1),$$

де $g(p)$ – функція кроку алгоритму p.

Для унеможливлення утворення комбінації вихідних сигналів, що можуть призвести до аварії, значення вектора $c_j(t_p)$ перевіряються на співпадіння зі значенням вектора $e_j(t_p)$, що запрограмований у матриці E заборонених станів:

$$\varepsilon_p = \overset{r}{\underset{x=1}{\vee}} \left[\overset{m}{\underset{j=1}{\wedge}} \left(c_{ijj} = e_{xj} \right) \right],$$

де індекси j і x пробігають значення від 1 до m і r відповідно.

Якщо значення булевої функції $\varepsilon = 1$, це означає, що на виході керуючого автомата з’явилась заборонена комбінація, при цьому видача даних сигналів на керований об’єкт блокується і його робота припиняється.

При $\varepsilon = 0$ керуючі впливи змінюють стан керованого об’єкта, який контролюється системою відповідних датчиків. У математичній моделі це знаходить своє відображення у зміні координат вектора станів $a(t_p)$. Оскільки зміна координат вектора $a(t_p)$ у реальному об'єкті керування може відбуватися не у відповідності до запрограмованого алгоритму, то постає необхідність безперервного порівняння станів детермінованих входів $a_j(t_p)$ (j = 1, 2, ..., k) з їх очікуваними значеннями A_{ij} (j = 1, 2, ... , k), записаними до i-го рядку матриці станів A. При цьому в залежності від значення ознаки операції F_p виконується логічна операція «І» чи «АБО», в залежності від того, яким чином повинен відбуватись перехід до наступного кроку алгоритму – при співпадінні всіх фактичних станів датчиків циклу з їх очікуваними значеннями (логічна операція І), або при наявності сигналу хоча б від одного з датчиків (логічна операція АБО), спрацювання яких очікується на p-му кроці програми:

$$\alpha_p = \begin{cases} \overset{k}{\underset{j=1}{\wedge}} \left(a_j(t_p) = A_{ij} \right), & при \quad F_p = 0; \\ \overset{k}{\underset{j=1}{\vee}} \left(a_j(t_p) = A_{ij} \right), & при \quad F_p = 1. \end{cases}$$

При $\alpha_p = 0$ номер рядка матриці програми не змінюється $I = g(p-1)$. Якщо ж булева функція α_p отримає значення істинності 1, то

$$I = g(p) = g(p-1) + 1, \text{ якщо } ST_p = 0,$$

де ST_p – ознака дозволу переходу. При $ST = 1$ перехід відбувається не на наступний крок алгоритму, а за адресою переходу, що визначається станом стохастичних входів $b_j(t_p)$.

Необхідно зазначити, що процес відпрацювання алгоритму полягає у послідовному проходженні рядків програми і лише в місцях розгалуження, номер рядка не збільшується на одиницю, а може прийняти будь яке значення адреси переходу, що визначається сигналами на стохастичних входах.

З літературних джерел [3] відомо, що всі алгоритми можна розділити на три великі класи:

– нерозгалужені алгоритми;

– алгоритми із розгалуженнями;

– алгоритми із вкладеними циклами.

Але дослідження всіх цих класів алгоритмів показує, що аналіз можливості переходу за адресою, що не дорівнює попередній адресі плюс 1, можна звести до перевірки наявності лише однієї ознаки – назвемо її ST (дозвіл переходу). При цьому в залежності від класу алгоритму ця ознака може нести в собі різний сенс, а саме кінець відпрацювання попередньої підпрограми – в алгоритмах із розгалуженнями; кінець відпрацювання тіла циклу – в алгоритмах із циклами.

Для формування адреси переходу в розгалужених програмах порівнюється стан стохастичних входів $b_j(t_p)$ з умовами B_{xj}, запрограмованими у матриці B, в результаті маємо булеву функцію β_p:

$$\beta_p = \bigvee_{x=1}^{n}\left[\bigwedge_{j=1}^{u}\left(b_j\left(t_p\right) = B_{xj}\right)\right],$$

де індекси j і x пробігають значення від 1 до u і n відповідно, тобто аналізується вся матриця B одночасно. При цьому у випадку істинності булевої функції β_p визначається той або інший рядок програми для нового $(p + 1)$-го кроку алгоритму:

$$I = g(p) = \left[d_p = f\left(b_p\right)\right], \text{ якщо } \left(\beta_p \wedge ST_p\right) \vee \operatorname{Int} l_p = 1,$$

де $d_p = f(b_p)$ – вектор адреси переходу на p-му кроці алгоритму.

При цьому необхідно врахувати значення додаткової умови ST_p та наявність переривання $\operatorname{Int}l_p$. $ST_p = 0$ означає, що на p-му кроці виконується попередня підпрограма і перехід до іншої підпрограми неможливий, тобто крок алгоритму визначається тільки станом детермінованих входів $a_j(t_p)$ і необхідно дочекатись закінчення відпрацювання попередньої підпрограми і появи $ST_p = 1$. Якщо на p-му кроці алгоритму виникає ситуація при якій необхідно терміново почати відпрацювання іншої підпрограми (наприклад підпрограми аварійної зупинки), то значення переривання $\operatorname{Int}l_p$ стає рівним 1 і перехід відбувається без урахування значення ST_p.

Далі на наступному $(p + 1)$-му кроці алгоритму керуючі впливи і очікувані стани визначаються $g(p)$-м рядком матриць A, C і цикл повторюється аналогічно до описаного вище.

Висновки

Таким чином в результаті розробки математичної моделі паралельного керуючого автомата з'являється можливість побудови програмованих логічних контролерів паралельної дії (ППЛК) із розширеними функціональними можливостями і підвищеною надійністю.

Література

1. Закревский А.Д. Параллельный автомат // Доклады АН БССР. – 1984. – Т. 28, № 8. – С. 717-719.

2. Фурман И.А. Научно-технические основы создания и промышленного применения параллельных логических контроллеров на программируемых БИС с матричной структурой: Дис. … докт. техн. наук: 05.13.05. – К., 1989. – 197 с.

3. Митник Ю.Ш., Хмельницкий А.С. Программирование и алгоритмические языки: Учебник для техникумов по специальности «ЭВМ, приборы и устройства» – 2-е изд., перераб. и доп. – М.: Машиностроение, 1984. – 368 с.

Надійшла до редакції 16.02.2006

Рецензент: д-р техн. наук, проф. В.А. Краснобаєв, Харківський національний технічний університет сільського господарства ім. Петра Василенка.