

ФОРМАЛІЗАЦІЯ СИНТЕЗУ АЛГОРИТМІЧНИХ МОДЕЛЕЙ СИСТЕМ КЕРУВАННЯ В УМОВАХ НЕВИЗНАЧЕНОСТІ

Актуальність теми

Масове використання систем керування (СК) викликає необхідність автоматизації процесу їх синтезу та оптимізації. Основою цього є різноманітні моделі. Оскільки СК включає апаратні та програмні засоби, то її модель має бути універсальною. Такою універсальною моделлю є алгоритмічна модель. Алгоритмічні моделі систем широко використовуються в різних галузях науки і техніки вже протягом значного проміжку часу. Алгоритмічні моделі є основою програмування [1], алгоритмічні моделі використовуються для опису процесів в інформаційних системах [2] і системах контролю та керування технологічними процесами [3]. Останнім часом поняття алгоритму узагальнене до алгоритмічної моделі і розповсюджене на системи вимірювання, контролю та керування. Таким чином, постає **проблема** проектування систем керування на основі алгоритмічної моделі.

Практично всі системи керування (СК) містять невизначеність в різному вигляді. В умовах невизначеності вказана проблема має певні особливості і наразі немає загальновизнаних підходів до її розв'язання. У більшості теоретичних задач оптимізації мова йде про постановки і методи розв'язання задач, що не містять невизначеностей. Алгоритмічна модель може бути основою проектування СК в умовах невизначеності.

Однак у наявних роботах розглядається обмежений клас задач і видів невизначеності при побудові алгоритмічних моделей. Отже створення відповідної методики формалізації синтезу алгоритмічних моделей для проектування систем керування в умовах невизначеності є важливою теоретичною і прикладною задачею роботи.

Для розв'язання поставлених задач необхідно сформулювати алгебраїчну систему алгоритмічних моделей (АМ) програмних та апаратних засобів СК, розробити методики перетворення АМ у

форму, що враховує невизначеність параметрів та початкових даних СК, розробити підходи до застосування АМ для оцінювання характеристик та оптимізації структури СК.

Задача використання алгоритмічної моделі як єдиної форми опису апаратної і програмної частин системи керування розглядається вже досить давно. Алгоритмічні моделі як апаратної, так і програмної частини систем керування, можуть бути представлені у алгебраїчній формі. Алгебраїчна форма моделі дозволяє здійснити перехід до узагальненої операторної моделі, яка враховує невизначеність початкових даних, вхідних впливів та структурних змін системи. Для відображення в алгоритмічній моделі операцій над функціями невизначеності введені означення операторів перетворення функцій [4].

Існує багато підходів до формалізації алгоритмічних моделей [5, 6].

Наприклад, апарат *логічних схем алгоритмів* (ЛСА) використовується для формалізованого опису інформаційно-вимірювальних систем (ІВС) [6]. У ІВС поєднуються апаратні засоби, починаючи від датчиків і кінчаючи пристроями видачі інформації, а також усі програми, як програми необхідні для керування роботою власне системи, так і ті, що дозволяють вирішувати у ІВС вимірювальні й обчислювальні задачі, а також керувати конкретним експериментом [7].

При розробці ЛСА передбачається, що вони повинні:

- описувати функціонування як апаратної, так і програмно-керованої частин СК;
- по можливості наглядно відобразити зміст перетворень, що описуються;
- описувати не тільки інформаційні перетворення, але і службові операції;
- слугувати основою для складання програм;
- описувати функціонування ВОС з різною деталізацією.

Однак у наявних роботах не розглядаються особливості застосування алгоритмічних моделей до аналізу і оптимізації систем в умовах невизначеності.

У формалізованому вигляді подамо означення алгоритмічної моделі за допомогою нотації Бекуса – Наура (наведені лише основні означення):

<алгоритмічна модель >	::= <ідентифікатор> ! A(B) < блок > A(E)
<ідентифікатор>	::= M <номер >
< блок >	::= <ідентифікатор> ! < пусто > ! < елемент > ! [< блок > < блок >]
< елемент >	::= < перетворення > ! < умова > ! < знак >
< перетворення >	::= E< номер > (< операція > ; < вхідні дані > ; < результати >)
< умова >	::= w <номер >(< логічний вираз >; < вхідні дані >) < знак переходу >
< знак переходу >	::= \lfloor < номер > ! \rfloor < номер > ! >>
< знак >	::= < знак переходу > ! << !
< операція >	::= < пусто > ! < тип операції >! < зміст операції >
< тип операції >	::= < I > ! < S > ! < C > ! <T> ! ...
< C >	::= + ! - ! * ! div ! log ...
< вхідні дані >	::= < пусто > ! (< ім'я >; <тип> ! <значення вхідного даного >)
<результати >	::= < пусто > ! (<ім'я>; <тип>)

Примітки: 1. Наведені лише основні означення.

2. Службові символи нотації: <, > – межі синтаксичної конструкції; ::= – “це”; ! – “або”; ... - “інші”.
3. \lfloor < номер > - перехід вперед; \rfloor < номер > - перехід назад.
4. < I > - інформаційні операції; < S > - вимірювальні операції; < C > - обчислювальні операції; <T>- операція передачі.

Формалізуємо перетворення алгоритмічної моделі у вигляді алгебраїчної системи:

$$AS = (AM, OP) \quad (1)$$

де AM – множина алгоритмічних моделей; OP – множина операцій над ними.

Базовою операцією алгебри алгоритмічних моделей, яка забезпечує процес синтезу, є операція підстановки

$$P\{\langle \text{блок1} \rangle, \langle \text{блок2} \rangle\},$$

що означає заміну блоку 1 блоком 2.

Підстановки повинні відповідати очевидним умовам

$$\langle \text{вхідні дані 1} \rangle = \langle \text{вхідні дані 2} \rangle$$

$$\langle \text{результати 1} \rangle \supseteq \langle \text{результати 2} \rangle$$

Послідовне виконання двох підстановок P_1 і P_2 назовемо добутком перестановок. Легко показати, що

- добуток перестановок має властивість асоціативності $P_1*(P_2*P_3)=(P_1*P_2)*P_3$,
- існує нейтральний елемент $1=P\{\langle \text{блок1} \rangle, \langle \text{блок1} \rangle\}$,
- існує обернений елемент $P^{-1}=P\{\langle \text{блок2} \rangle, \langle \text{блок1} \rangle\}$.

Така сукупність властивостей підстановок є підставою для твердження, що підстановки утворюють нескінченну групу [8]. Цей висновок є теоретичним підґрунтям використання алгебраїчних методів при синтезі СК на основі алгоритмічних моделей.

Операція підстановки еквівалентна множині операцій OP , яка складається з двох елементів:

- $paste(B, n1, n2)$ - вставка блока B в алгоритмічну модель між елементами з номерами $n1$ і $n2$;
- $cut(n1, n2)$ - вирізання блока з алгоритмічної моделі між елементами з номерами $n1$ і $n2$,

а також понять одиничної операції 1 , яка не змінює моделі, і зворотної операції op^{-1} , яка задовольняє умові

$$op^{-1}(op(M)) = M. \quad (2)$$

Еквівалентними перетвореннями алгоритмічної моделі будемо називати таку послідовність операцій над моделлю, яка не змінює змісту результатів роботи системи (хоча можуть змінювати якісні показники як результатів, так і самої системи).

Розв'язком задачі синтезу моделі СК є послідовність підстановок, яка деталізує і оптимізує базову модель $M=A(B)EO(op_1; X; Y)A(E)$.

Оптимізація в процесі синтезу здійснюється на основі вартісної функції $g([P_{11}, P_{12}], [P_{21}, P_{22}])$, де P_{11} – множина послідовних операцій; P_{12} – множина паралельних операцій; P_{21} – множина апаратно реалізованих операцій; P_{22} – множина програмно реалізованих операцій. Вартісна функція повинна відповідати умовам:

- вартісна функція паралельно з'єднаних підсистем

$$g_{nap} = \sum_i g_{P_{12}i} + c_t \max_i (\sum_{P_{11}} T_i), \quad (3)$$

де $g_{P_{12}i}$ - вартість апаратних засобів i -ї підсистеми,

T_i - час виконання операції i -ю підсистемою, яка здійснює послідовність операцій P_{11} ,

c_t - вартість витрат часу;

- вартісна функція послідовно з'єднаних підсистем

$$g_{noc} = \sum_i g_{P_{12}i} + c_t \sum_i T_i. \quad (4)$$

Вираз вартісної функції отримується з алгоритмічної моделі СК.

Оптимізація здійснюється шляхом зміни двох показників: програмна або апаратна реалізація операцій, послідовне або паралельне виконання операцій.

Отже, задача оптимізації структури системи керування може бути розглянута як задача знаходження оптимуму функції критерію оптимальності на множині значень параметрів, для чого необхідно отримати алгоритмічну модель системи керування.

При розв'язанні задачі оптимального розподілу операцій на апаратні і програмні здійснюються еквівалентні перетворення моделі, які полягають у заміні частини апаратних операцій на програмні та додаванні інтерфейсних перетворень або навпаки.

Заміна апаратних перетворень на програмні і навпаки здійснюється на підставі тверджень

$$1. E_0(op_H, X_H, Y_H) = E_1(I_1, X_H, X_S) E_2(op_S, X_S, Y_S) E_3(I_2, Y_S, Y_H)$$

$$2. E_0(op_S, X_S, Y_S) = E_1(I_2, X_S, X_H) E_2(op_H, X_H, Y_H) E_3(I_1, Y_H, Y_S)$$

де H – позначення апаратно-орієнтованих операцій і форми даних; S – позначення програмно-орієнтованих операцій і форми даних; I_1 – апаратно-програмне інтерфейсне перетворення; I_2 – програмно-апаратне інтерфейсне перетворення.

Отже, заміна апаратних перетворень $[n1 - n2]$ на програмні здійснюється за допомогою операцій

$$cut(n1, n2) \text{ paste}(E_1(I_1, X_H, X_S) E_2(op_S, X_S, Y_S) E_3(I_2, Y_S, Y_H)), \\ prev(n1), next(n2) \quad (5)$$

і аналогічно при зворотному перетворенні.

При розпаралелюванні виконання операцій до алгоритмічної моделі додаються операції синхронізації і збирання даних, що паралельно надходять.

Заміна послідовних операцій на паралельні здійснюється на підставі твердження

$$3. E_1(op_1, X1, Y) E_2(op_2, X2, Y) = \| [E_3(I_3, (X1, X2), X1) E_1(op_1, X1, Y1)] [E_3(I_3, (X1, X2), X2) E_2(op_2, X2, Y2)] E_4(I_4, (Y1, Y2), Y) \quad (6)$$

де I_3 – операція розподілення вхідних даних; I_4 – операція об'єднання результатів.

Можливі перетворення апаратних операцій на програмні задаються базою знань. При заміні частини апаратних операцій на програмні до алгоритмічної моделі додаються операції інтерфейсних

перетворень. При розпаралелюванні виконання операцій до алгоритмічної моделі додаються операції синхронізації і збирання даних, що паралельно надходять. Процедура розпаралелювання використовує ГПС – граф перетворення сигналів і ГЗ – граф залежностей [9].

Методика оптимізації СК на основі алгоритмічної моделі передбачає:

- аналіз залежностей у послідовностях перетворень і виділення незалежних послідовностей;
- розпаралелювання перетворень на основі (6);
- апаратна підтримка паралельності на основі операції (5);
- оцінка витрат та вибір найкращого варіанту на основі критеріїв (3) і (4);
- зміна структури системи у відповідності до обраного варіанту.

Процедура розпаралелювання використовує ГПС – граф перетворення сигналів і ГЗ – граф залежностей. Для формалізації процедури в алгебрі AS визначаються відносини залежності \otimes :

$E_1(op_1, X1, Y1) \otimes E_2(op_2, X2, Y2)$ якщо $(X1 \otimes X2) \cup (X1 \otimes Y2) \cup (X2 \otimes Y1)$

Для відносин залежності виконується властивість транзитивності.

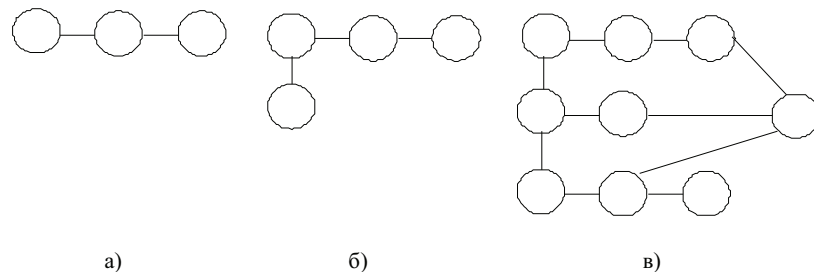


Рисунок 1 – Граф перетворень і залежностей

Аналіз залежностей може бути здійснений за допомогою динамічної структури типу двовимірний зв'язаний список, який формується за алгоритмом рис.2.

Отриманий таким способом граф перетворень і залежностей відтворює максимально можливе розпаралелювання.

При синтезі системи керування необхідно враховувати невизначеність та її вигляд. Невизначеність може походити з різних джерел, відповідно по-різному описуватися і враховуватися у технічних системах. Реальні інтелектуальні системи, як і людина, повинні ефективно функціонувати в умовах одночасного існування

практично всіх видів невизначеності. Для технічних систем особливо поширеною є комбінація стохастичної та нечіткої невизначеності.

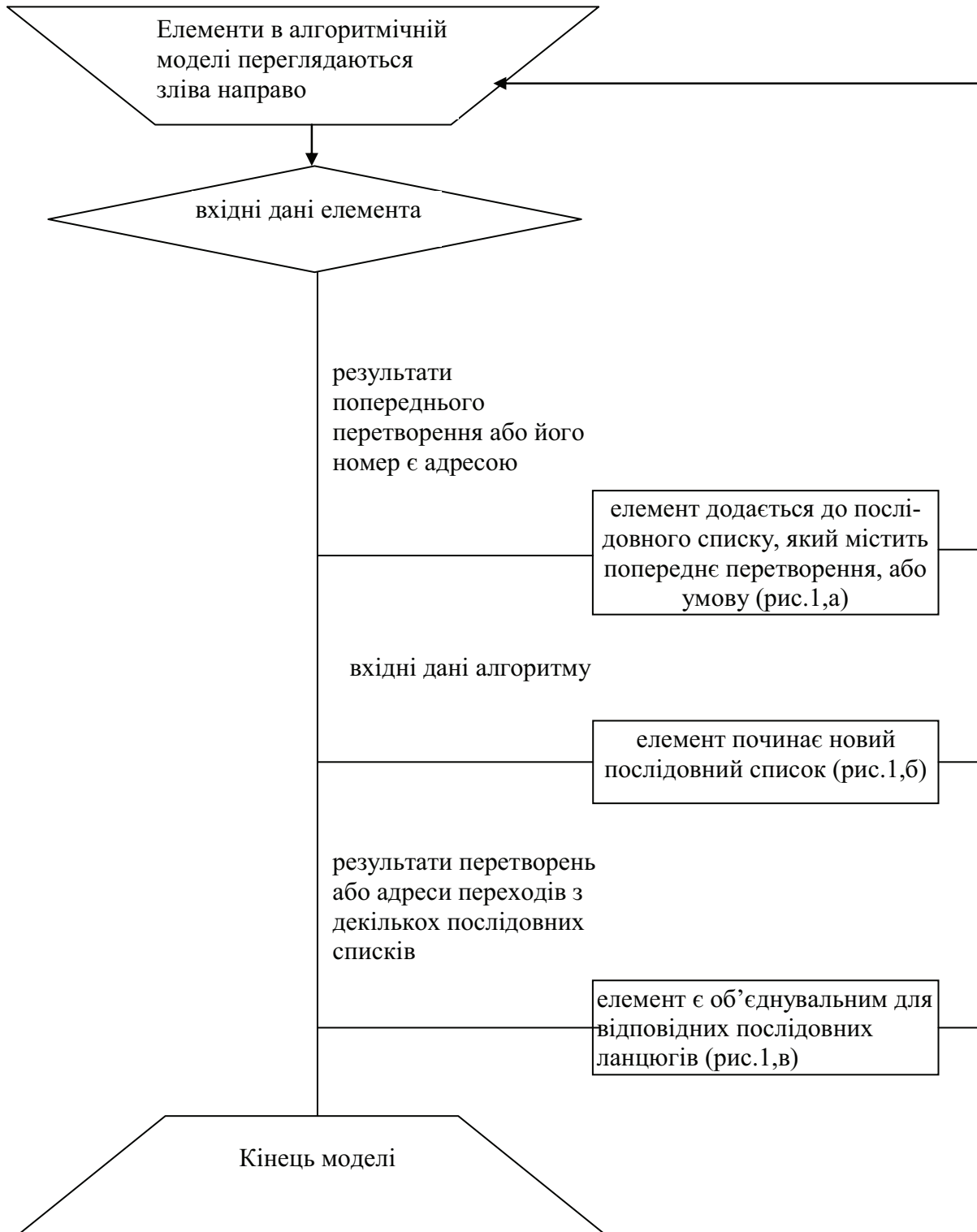


Рисунок 2 – Алгоритм синтезу структури моделі

Джерелом невизначеності може бути недостатнє знання предметної області чи інформації про конкретну ситуацію, недостача точної інформації про значення даних, невизначеність цілей. Існуючі

методи моделювання розглядають функціонування систем контролю та керування в умовах невизначеності стохастичного [10] чи нечіткого [11] типу. Маються також роботи по нечітким і стохастичним алгоритмах. Запропонований також метод узагальнення невизначеності [12].

Синтез моделі СК в умовах невизначеності здійснюється на основі методу узагальнюючих функцій [12].

Під узагальнюючою функцією розуміють додатно визначену функцію на проміжку можливих значень аргументу, що позначається $\beta(x)$ і характеризує можливість π або ймовірність P прийняття аргументом значення з певного інтервалу $[x_1, x_2]$, $x_1 \in B$, $x_2 \in B$, за правилами:

$$p = \frac{\int_{x_1}^{x_2} d[\beta(x)]}{\int_B d[\beta(x)]}, \quad \pi = \frac{\int_{x_1}^{x_2} d[\beta(x)]}{\max_B \int_{[x_{i-1}, x_i]} d[\beta(x)]} \quad (7)$$

де $x_{i-1}, x_i \in B$, $i = \overline{1, n}$, n – кількість інтервалів розбиття B .

Метод узагальнюючих функцій, який враховує невизначеність різного типу, дозволяє представити оператори параметричного типу (оператори присвоювання) і оператори структурного типу (оператори переходу за результатами перевірки умов) за допомогою узагальнюючих функцій і визначених для них операторних перетворень. Система узагальнюючих функцій [12] складається з правил утворення формул, правил переходу від формальних систем чітких чисел, випадкових величин, нечітких чисел до системи узагальнюючих функцій і назад. Для узагальнюючої функції визначені також правила узагальнення математичних операцій. Всі операції поділені на три групи: нелінійні унарні, нелінійні бінарні, інтегро-диференційні. Узагальнена математична операція виконується за допомогою інтегрального оператора, ядро якого визначається типом операції.

$$\beta_Y(y) = \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} \beta_X(\bar{x}) \psi(\bar{x}, y, F, Q) d\bar{x}, \quad (8)$$

де ψ - ядро оператора, яке задовольняє умові

$$\int_{-\infty}^{+\infty} \psi(\bar{x}, y, F, Q) d\bar{x} = 1,$$

де F і Q – характеристики операції, яка виконується; n – кратність інтегрування, яка залежить від розмірності вектора \bar{x} і характеристик операцій F і Q .

З метою врахування невизначеності в кожному <перетворенні> алгоритмічної моделі виконується заміна <операції> на відповідне ядро інтегрального оператора, а <вхідні дані > і < результати > подаються узагальнюючими функціями.

Висновки

Розроблено методику формалізації синтезу алгоритмічних моделей для проектування систем керування в умовах невизначеності на основі методу узагальнюючих функцій. Запропонований підхід до формалізації операцій над алгоритмічною моделлю систем керування дозволяє здійснити оптимізацію системи на основі розпаралелювання перетворень та вибору програмного або апаратного способу їх реалізації.

ЛИТЕРАТУРА

1. Н.Вирт. Алгоритмы + структуры данных = программы: Пер. с англ. Л.Ю. Иоффе. – М.: Мир, 1985. – 406с.
2. Цапенко М.П. Измерительные информационные системы: Структуры и алгоритмы, системотехническое проектирование. Учебн. Пособие для вузов. – 2-е. изд. перераб. и доп. – М.:Энергоатомиздат, 1985. – 439с.
3. Ротштейн А.П. Интеллектуальные технологии идентификации. – Винница: Универсум, 2000. – 400 с.
4. Дубовой В.М., Никитенко О.Д. Визначення вимог до структури підсистеми керування вимірювально-обчислювальної системи. Вісник Хмельницького національного університету – 2005.- №4 Ч.1, Т.1 (68), с.115-118.
5. Глушков В.М., Цейтлин Г.Е., Ющенко В.М. – Алгебра. Языки. Программирование./ АНУССР, Ин-т киберн. Им. В.М.Глушкова. – 3 изд.. К: Наукова думка, 1989. - 376с.

6. Цапенко М.П. Содержательные логические схемы алгоритмов измерительных систем. – Измерения. Контроль. Автоматизация, 1982, №4, с.3-8.
7. V. Dubovoy, O. Nikitenko Algorithmic models of systems in conditions of uncertainty //Матеріали IV міжнародної конференції „Інтернет-Освіта-Наука” (ІОН-2004). Том 2, – Вінниця: Універсум. - 2004. – С.538-541
8. Александров П.С. Введение в теорию групп. – М.: Наука, 1980, 144 с.
9. Кун С.Ю. Матричные процессоры на СБИС: Пер. с англ. – М.: Мир, 1991.–672с.
10. Пугачов В.С. и др. Основы статистической теории автоматических систем. – М.: Машиностроение, 1974. – 400с.
11. Нечеткие множества в моделях управления и искусственного интеллекта/ Под ред. Д.А.Поспелова. – М.: Наука. Гл. ред. физ.-мат.лит., 1986. – 312с.
12. Глонь О.В., Дубовой В.М. Моделювання систем керування в умовах невизначеності. Монографія. – Вінниця: УНІВЕРСУМ-Вінниця, 2004.– 170 с.